НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО

Факультет інформатики та обчислювальної техніки
(назва факультету, інституту)

Кафедра автоматизованих систем обробки інформації і управління
(назва кафедри)

"На правах рукопису"
УДК *004.023*

«До захисту допущено»
Завідувач кафедри

_____ О.А.Павлов
(підпис) (ініціали, прізвище)

“____” _____ 20*18* р.

МАГІСТЕРСЬКА ДИСЕРТАЦІЯ

на здобуття ступеня магістра

за спеціальністю *126 Інформаційні системи та технології*
(код та назва спеціальності)

ОПП *Інформаційні управляючі системи та технології*
(код та назва спеціалізації)

на тему: *Програмно-аналітичний комплекс поділу користувачів соціальної мережі на групи за інтересами*

Виконав: студент *VI* курсу групи ІС-72мп
(шифр групи)

Булгар Максим Миколайович _____
(прізвище, ім’я, по батькові) (підпис)

Науковий керівник *доц., к.т.н., доц. Жаріков Е.В.* _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант *к.т.н., доц. Жданова О.Г.* _____
(науковий ступінь, вчене звання, прізвище, ініціали) (підпис)

Рецензент _____ _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.

Студент _____
(підпис)

Київ – 2018

РЕФЕРАТ

Магістерська дисертація: 100 с., 15 рис., 34 табл., 1 додаток, 31 джерел.

Актуальність. Сьогодні Інтернет є основним джерелом отримання або поширення інформації. Більшість людей використовує Інтернет для того, щоб проводити час у соціальних мережах. Кожна людина може спробувати себе у ролі репортера, і поділитись важливими новинами, або поширити свою думку серед певної аудиторії і знайти однодумців/послідовників.

На початок 2018 року його аудиторія становила 336 мільйонів людей [1]. Користувачі спілкуються короткими повідомленнями (до 280 символів) які називаються твітами. Це дуже зручно, так як можна отримувати дуже великий об’єм інформації, витрачаючи на це не дуже велику кількість часу.

Саме тому є можливою розробка системи, яка зможе групувати користувачів схожих за інтересами беручи за основу текстову складову їх поведінки у соціальній мережі Twitter.

Зв'язок роботи з науковими програмами, планами, темами. Робота виконувалась на кафедрі автоматизованих систем обробки інформації та управління Національного технічного університету України «Київський політехнічний інститут ім. Ігоря Сікорського» в рамках теми «Методи та технології високопродуктивних обчислень та обробки надвеликих масивів даних». Державний реєстраційний номер 0117U000924.

Мета дослідження – підвищення якості поділу користувачів соціальної мережі на групи за рахунок аналізу текстової складової їх поведінки.

Для досягнення мети необхідно виконати наступні **завдання**:

– проаналізувати дані, які можна отримати про користувача в соціальній мережі Twitter;

– обрати критерії за якими порівнюються користувачі;

– проаналізувати методи та засоби кластеризації великих даних;

– дослідити способи аналізу поведінки користувача за текстовою складовою;

– реалізувати програмно аналітичний комплекс поділу користувачів соціальної мережі на групи за інтересами.

Об’єкт дослідження – процес поділу користувачів на групи, в залежності від схожості їх інтересів.

Предмет дослідження – методи кластеризацій користувачів соціальної мережі на основі їх текстової складової поведінки.

Наукова новизна отриманих результатів полягає в аналізі існуючих методів кластеризації. Також було проаналізовано способи аналізу текстової складової, так як одним із критеріїв порівняння схожості було взято тематика блогінгу.

Публікації. Булгар М.М. Кластеризація користувачів за їх інтересами / М.М. Булгар // МОДС. 2018. С. 28-29.

Булгар М.М. Спосіб кластеризації користувачів соціальної мережі Twitter / М.М. Булгар // ІСТУ. 2018. С. 28-32.

КЛАСТЕРИЗАЦІЯ, СОЦІАЛЬНІ МЕРЕЖІ, ВЕЛИКІ ДАНІ, TWITTER, ТВІТ, АНАЛІЗ

ABSTRACT

Master's dissertation: 92 p., 15 fig., 34 tabl., 1 appendix, 31 sources.

Topicality. Today, the Internet is the main channel for receiving or distributing information. Most people use the internet to stay in social networks. Everyone can feel like a true reporter, and share important news, or spread his thoughts to the masses, and find like-minded people / followers. You can also be a simple reader and draw information from other people.

Among all social networks, Twitter has its niche. At the beginning of 2018, its audience was 336 million people. Users communicate with short messages (up to 280 characters) called tweets. This is very convenient, since you can get a very large amount of information, spending it on not too much time.

That is why it is possible to develop a system that can group interest-based users based on the textual component of their behavior in the social network Twitter.

Relationship with academic programs, plans, themes. Work performed at the Department of ASOIU at the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute" within the topic “Methods and technologies of high performance computing and performing of big data”. Governments register number 0117U000924.

The aim of the research is improving the quality of sharing social network users into groups by analyzing the textual component of their behavior.

To achieve the goal need to accomplish the following **tasks**:

- analyze the data you can get about the user in the social network twitter;
- choose the criteria by which users are compared;
- analyze methods and means of clustering of large data;
- explore ways to analyze user behavior over a text component;
- to implement software analytical system of division of social network users into interest groups.

Object of research - the process of grouping users into groups of common interests.

Subject of research - methods of clustering users of the social network based on a large number of data on their activities.Scientific novelty of the obtained results.

The scientific novelty of the results - to analyze the existing methods of clustering. The methods of analysis of the text component were also analyzed, since one of the criteria for comparing similarity was the topic of blogging.

Published works.

Bulgar M.M. Clustering users through their interests / M.M. Bulgar // MODS. 2018. pp. 28-29.

Bulgar M.M. Methods of clusterization the users of the social network twitter / M.M. Bulgar // ISMT. 2018. pp. 28-32.

CLUSTERING, SOCIAL NETWORK, BIG DATA, TWITTER, TWEET, ANALYS

ЗМІСТ

1 ПРОЕКТНІ РІШЕННЯ З РОЗРОБКИ СИСТЕМИ ПРОГРАМНО АНАЛІТИЧНИЙ КОМПЛЕКС ПОДІЛУ КОРИСТУВАЧІВ СОЦІАЛЬНОЇ МЕРЕЖІ НА ГРУПИ ЗА ІНТЕРЕСАМИ

1.1 Опис бізнес-процесів

Інтернет є основним джерелом отримання інформації. Більшість людей використовує Інтернет для того, щоб проводити час у соціальних мережах. Кожна людина може спробувати себе у ролі репортера і поділитись важливими новинами, або поширити свою думку серед певної аудиторії і знайти однодумців/послідовників.

На початок 2018 року його аудиторія становила 336 мільйонів людей [1]. Користувачі спілкуються короткими повідомленнями (до 280 символів) які називаються твітами. Це дуже зручно, так як можна отримувати дуже великий об’єм інформації, витрачаючи на це не дуже велику кількість часу.

Люди надсилають твіти з великої кількості причин:

– особливість інтересів;
– бажання поділитись чимось;
– важливі новини.

Завдяки особливості твітів, користувач отримує всі важливі для нього новини у зручний спосіб. І завдяки тому, що аудиторія та кількість даних росте, можна провести різні аналізи цієї інформації.

Також розглянемо інші соціальні мережі.

Facebook – це соціальна мережа, у якій користувач має можливість:

– бути підписаним на деякі групи;
– додавати у список друзів інших користувачів;
– створювати пости з різною прикріпленою інформацією.

Недоліком цієї соціальної мережі є те, що їхня API (application programming interface) надає недостатню кількість інформації. Цю інформацію

важко та незручно аналізувати, так як її стандартизація мінімальна. Набула популярності в Україні після закриття російських соціальниїх мереж.

Instagram. Основною особливістю є те, що користувачі діляться своїми фотографіями, де можуть прикріплювати геолокацію, хештеги та інформацію подібну до цієї. Аналізувати щось в цій системі важко, так як дані не дуже підходять для цього.

На противагу Twitter, Facebook, є значно менш публічним середовищем. Тут спілкування переважно відбувається в приватних або напівприватних налаштуваннях. Зв'язки дружби між обліковими записами вимагають взаємності, а механізми швидкого збору громадськості навколо загальних цікавих тем, таких як хеш-тег, доступні, але рідко використовуються.

Таким чином, для аналізу обрана соціальна мережа Twitter. Twitter надає коротку, змістовну інформацію. Обмеженням у довжині повідомлення соціальна мережа примушує своїх користувачів висловлюватись стисло та змістовно. Тому кожен твіт несе в собі велику кількість інформації, проаналізувавши яку можна зрозуміти чим цікавиться людина, який її настрій а також інформацію схожу на цю.

Треба зазначити, що зараз соціальні мережі стають менш популярні, так як користувачі все більше переходять на месенджери, де вони можуть спілкуватись, не хвилюючись за те, що про них збирається інформація.

1.1.1 Опис процесу діяльності

Twitter в даний час широко визнається як особливо важлива глобальна платформа для відкритого спілкування.

Важливим є те, що переважна більшість облікових записів Twitter та їх твітів є загальнодоступними для всіх інших користувачів і навіть для незареєстрованих відвідувачів. Дуже короткий формат повідомлень дає змогу зробити розмову між користувачами таку, що нагадує усне, а не письмове спілкування.

Зрозуміло, що публічність більшості повідомлень у Twitter не означає, що твіти звичайних користувачів регулярно охоплюють велику аудиторію. Дійсно, більшість всіх твітів, швидше за все, бачитимуть лише декілька інших користувачів, а сам Twitter залишається платформою для невеликих груп людей яких пов’язують загальні інтереси.

Проте загальна доступність за умовчанням для більшості облікових записів Twitter та їх повідомлень, відсутність перешкод для доступу до цих облікових записів, а також відносна легкість, з якою окремі публікації можуть бути поширені через мережу (через retweeting) та через інші платформи (за допомогою вбудовування) неодноразово надавали швидкі інформаційні каскади, які посилюють видимість окремих твітів набагато більше, ніж вихідна аудиторія їх послідовників, і навіть значно випереджає саму базу користувачів Twitter.

Висока видкість поширення інформації через Twitter найбільш поширена для твітів політиків, знаменитостей та інших важливих діячів у суспільному житті. Але також може зустрітись, коли звичайні користувачі опиняються в надзвичайних обставинах, наприклад, на місці великої кризової події.

Якщо проаналізувати вибори 2008 року у США, то можна побачити як соціальні мережі були використані у політичній боротьбі. Деякі з аналітиків заявляли, що це була одна з вирішуючих частин його кампанії [2].

У цьому контексті загальним колективним терміном для мережевого комунікативного середовища Twitter є "Twittersphere". Twitter використовується в діапазоні від соціальної взаємодії з дітьми та культури до напружених політичних дебатів. Проте ясно, що термін "Twittersphere" чітко вказує на вроджене розуміння користувачем Twitter (і дослідників Twitter), розуміння простору соціальних мереж та взаємодії Twitter як середовища з відмінними внутрішніми структурами, які спрямовують потік інформації та спілкування між людьми та групами соціуму, і, таким чином, в кінцевому підсумку

представляють сукупність відносин. Такі структури мають потенціал суттєво вплинути на новини та інформацію, доступну користувачам Twitter.

Існують деякі дослідження, які вивчають, наприклад, внутрішні структури в окремих країнах або для конкретних областей інтересу. Структури, створені протягом більш тривалого періоду часу шляхом поступового накопичення користувачів і зв’язаних з ними твітів. Розвиток таких структур може бути пов'язано, зокрема, із сумішшю існуючих особистих зв'язків за межами Twitter, організовані кампанії по інтересам та/або вбудовані можливості платформи, які надають рекомендації за ким слідкувати.

Всередині системи головним процесом діяльності є процес створення користувачем заявки та її подальший аналіз. Користувач створює заявку на аналіз та прікріплює до неї файл. У файлі міститься інформація про те, яких користувачів буде проаналізовано. Для запуску заявки в обробку необхідна модерація адміністратора. Це зроблено для того, щоб мати змогу відхилити заявки, які не задовільняють вимогам.

1.1.2 Актори і функції

Наведемо список акторів і функцій системи у вигляді діаграми варіантів використання. Як видно з діаграми (рисунок 1.1) в системі існують 2 актори – **користувач** та **адміністратор.**

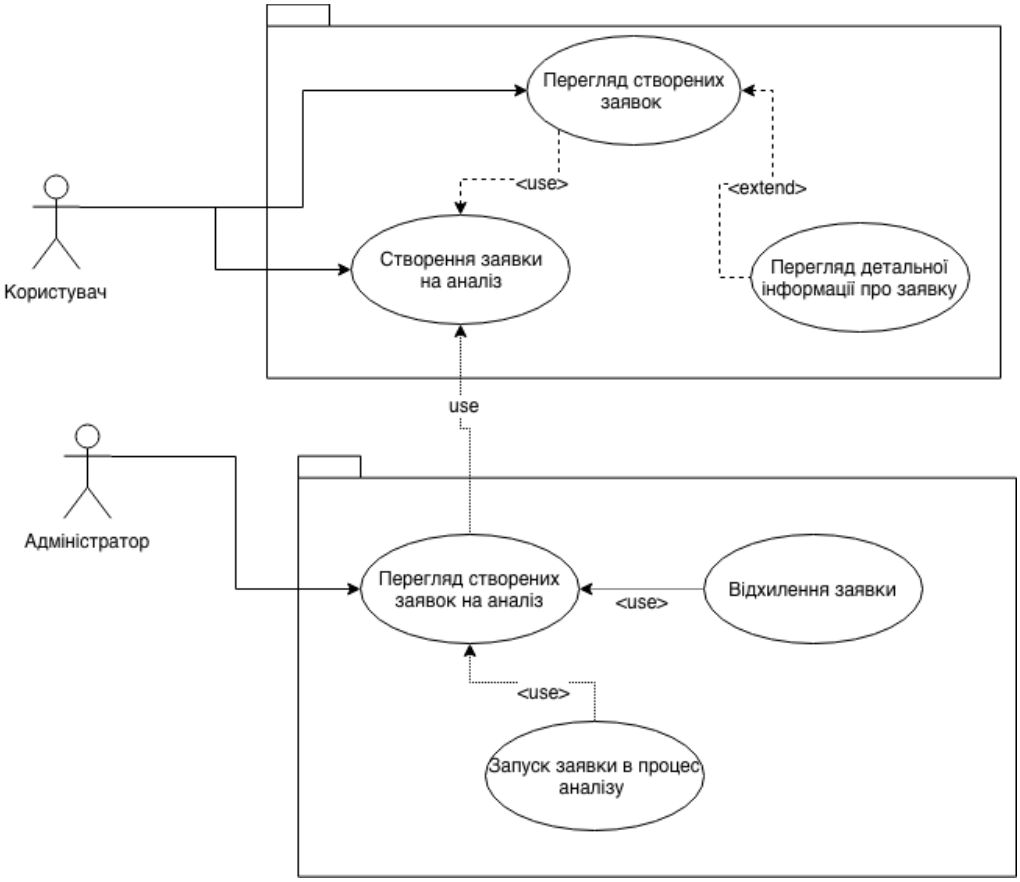

Рисунок 1.1 – Діаграма використання

Кожен з акторів може взаємодіяти з системою наступним чином.

Адміністратор:

– керування створеними заявками.

Користувач:

– створення заявок;
– перегляд створених заявок;
– перегляд детальної інформації про виконані заявки.

В таблиці 1.1 наведено розписані складові варіантів використання актора Адміністратор.

Таблиця 1.1 – Варіанти використання для актора Адміністратор

Актор	**Назва варіанту**	**Складові варіанта**
Адміністратор	*Перегляд створених заявок*	*Запуск заявки в процес аналізу*
		Відхилення заявки

В таблиці 1.2 наведено розписані складові варіантів використання актора Користувач.

Таблиця 1.2 – Варіанти використання для актора Користувач

Актор	**Назва варіанту**	**Складові варіанта**
Користувач	*Створення заявки*	*Заповнення інформації необхідної для створення заявки*
	Управління заявками	*Перегляд створених заявок*
	Перегляд створених заявок	*Перегляд детальної інформації про заявку*
		Видалення заявки
	Перегляд детальної інформації про заявку	*Редагування інформації про заявку*

Опишемо кожний з варіантів використання більш детально:

Перегляд створених заявок – адміністратор займається управлінням заявками у системі. Для цього він заходить до панелі адміністратора, і там може виконувати необхідні йому дії над заявкою.

Запуск заявки в процес аналізу – адміністратор попередньо переглянувши інформацію залишину користувачем в заявці, упевнюється у її правильності та запускає заявку в процес аналізу.

Відхилення заявки – адміністратор перевіривши інформацію про заявку, відхиляє її, якщо ця заявка не задовільняє вимогам.

Створення заявки – користувач має можливість створити заявку та відправити її на розгляд адміністратору. Для того, щоб створити заявку на аналіз користувач має заповнити причину, чому він хоче виконати аналіз, прикріплює файл з інформацією про користувачів, яких потрібно проаналізувати. Після заповнення всіє інформації користувач підтверджує створення заявки.

Перегляд детальної інформації про заявку – користувач має можливість переглянути детальну інформацію про заявку, внести деякі зміни до неї. При необхідності, також, може видалити заявку з черги на модерацію. Якщо заявка виконана, до неї буде прикріплено файл з результатами аналізу.

1.1.3 Структура бізнес-процесів

Основними процесами діяльності всередині системи є:

– створення нової заявки на аналіз;
– запуск заявки в роботу;
– перегляд результатів аналізу.

Для роботи з системою користувачі мають зареєструватися, щоб отримати доступ до особистого кабінету. Після реєстрації користувач потрапляє в особистий кабінет.

В особистому кабінеті користувач може створити заявку на аналіз. Для цього він переходить у відповідний розділ особистого кабінету. Далі користувач має заповнити необхідну інформацію для створення заявки (ціль аналізу, файл з вхідними даними). Після цього натискає кнопку підтвердження створення заявки (рисунок 1.2).

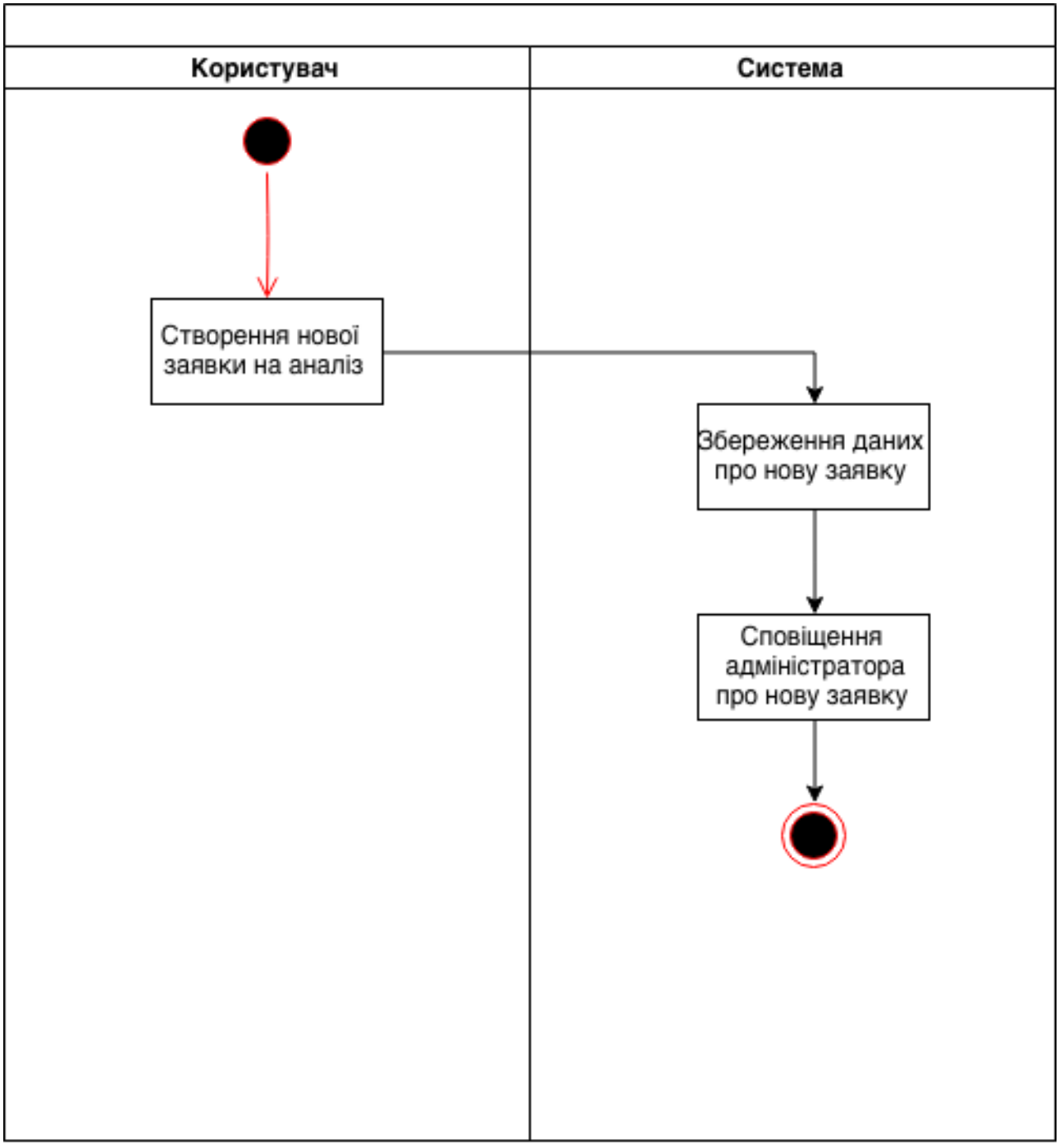

Рисунок 1.2 – Діаграма діяльності “Створення заявки на аналіз”

Далі створена заявка потрапляє до списку усіх заявок у кабінеті адміністратора. Адміністратор переглядає заявку, перевіряє коректність вхідних

даних для аналізу, і далі, якщо заявка задовільняє вимогам, запускає її в процес аналізу (рисунок 1.3). Якщо дані не задовільняють вимогам, адміністратор може відхилити заявку (рисунок 1.3).

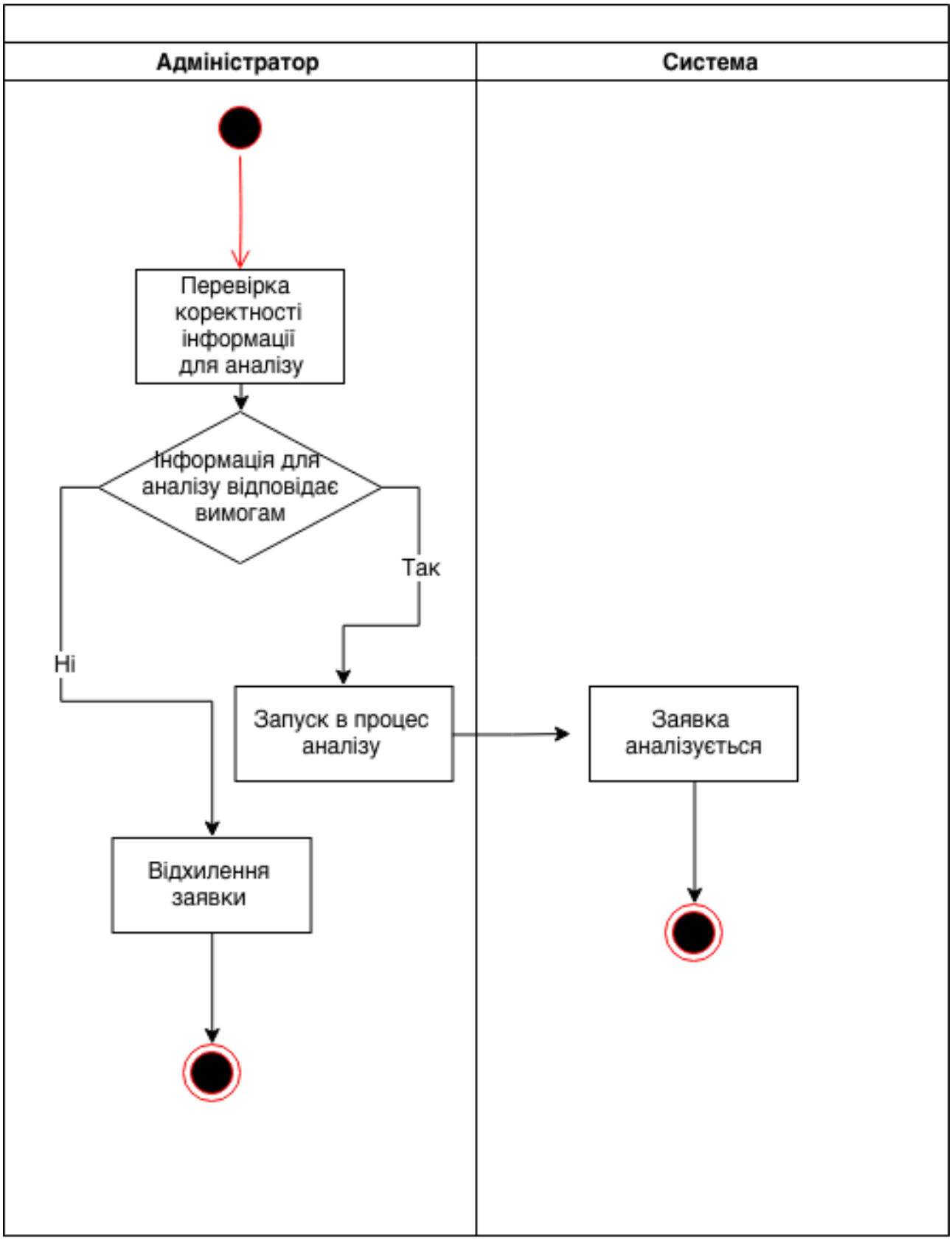

Рисунок 1.3 – Діаграма діяльності “Запуск заявки в обробку”

Після того, як аналіз заявки виконаний, користувачу приходить електронний лист про те, що заявка виконана, і що він може переглянути результат аналізу.

1.2 Опис постановки задачі

Система призначена для поділу користувачів соціальної мережі Twitter на групи в залежності від схожості інтересів цих користувачів.

Ціль створення: Отримання даних вподобання користувачів мережі Twitter та їх схожості між собою.

Для досягнення мети необхідно виконати наступні **завдання**:

– проаналізувати дані, які можна отримати про користувача в соціальній мережі Twitter;

– обрати критерії за якими порівнюються користувачі;

– проаналізувати методи та засоби кластеризації великих даних;

– дослідити способи аналізу поведінки користувача за текстовою складовою;

– реалізувати програмно аналітичний комплекс поділу користувачів соціальної мережі на групи за інтересами.

Задачі системи. Для реалізації програмно аналітичного комплексу поділу користувачів система повинна вирішувати такі задачі:

– створення заявки;

– керування заявками користувачем;

– керування заявками адміністратором;

– виконання аналізу інформації з соціальної мережі Twitter шляхом автоматизованого розбиття користувачів на кластери за їх вподобаннями.

1.3 Рішення з інформаційного забезпечення

При розробці даної системи була використана реляційна СУБД PostgreSQL. Структура бази даних описана у вигляді ER-діаграми (рисунок 1.4).

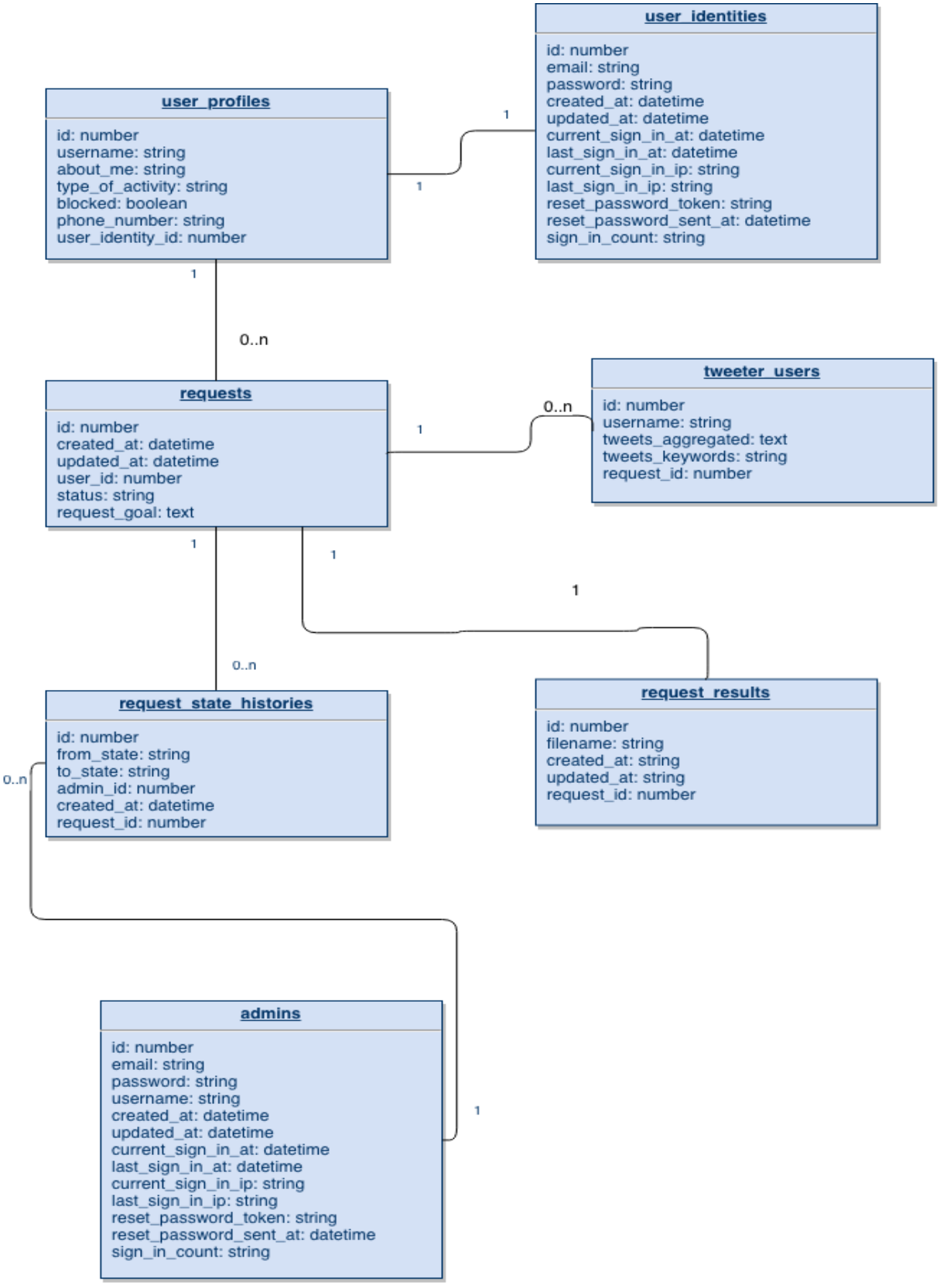

Рисунок 1.4 – ER-діаграма бази даних

Наведемо опис таблиць, що використовуються для зберігання даних всередині системи та відображені на вищезазначеній діаграмі (таблиця 1.3).

Таблиця 1.3 – Таблиці бази даних

Назва таблиці	**Опис**
admins	Інформація про адміністратора системи
user_profiles	Інформація про користувача
user_identities	Інформація про дані користувачі необхідні для авторизації
requests	Інформація про запити на аналіз
request_results	Посилання на файл за результатами аналізу, а також базова інформація про аналіз
request_state_histories	Інформація про зміну статусів заявки
tweeter_users	Інформація про користувачів Twitter, які вже були проаналізовані

В таблиці admins зберігається інформація про адміністраторів, та інформація, яка необхідна для їх авторизації у системі (таблиця 1.4).

Таблиця 1.4 – Поля таблиці admins

Параметр	**Тип поля**	**Короткий опис**	**Обмеження**
1	**2**	**3**	**4**
id	integer	ID адміністратора	Не може бути порожнім. Внутрішній ключ
email	string	Email адміністратора	Не може бути порожнім
password	string	Пароль адміністратора	Не може бути порожнім
username	string	Ім’я адміністратора в системі	
created_at	datetime	Дата створення запису	Не може бути порожнім
updated_at	datetime	Дата останнього оновлення запису	
current_sign_in_at	datetime	Дата поточного входу в систему	
last_sign_in_at	datetime	Дата останнього входу в систему	

Продовження Таблиці 1.4

1	2	3	4
current_sign_in_ip	string	IP-адреса поточного входу в систему	
last_sign_in_ip	string	IP-адреса останнього входу в систему	
reset_password_token	string	Ключ для відновлення паролю	
reset_password_sent_at	datetime	Дата останнього відновлення паролю	
sign_in_count	integer	Кількість входів в систему	

В таблиці user_profiles зберігається інформація про профілі користувачів (таблиця 1.5).

Таблиця 1.5 – Поля таблиці user_profiles

Параметр	**Тип поля**	**Короткий опис**	**Обмеження**
id	integer	ID профілю користувача	Не може бути порожнім. Внутрішній ключ
username	string	Ім’я користувача в системі	Не може бути порожнім
about_me	string	Описа типу зайнятості користувача	Не може бути порожнім
type_of_ activity	string	Опис основної зайнятості користувача	
blocked	boolean	Чи заблокований користувач у системі	Не може бути порожнім
phone_ number	string	Дата останнього оновлення запису	
user_ identity_id	integer	ID сутності авторизації користувача	Не може бути порожнім. Зовнішній ключ.

В таблиці user_identities зберігається інформація, яка необхідна для авторизації користувачів у системі (таблиця 1.6).

Таблиця 1.6 – Поля таблиці user_identities

Параметр	**Тип поля**	**Короткий опис**	**Обмеження**
1	**2**	**3**	**4**
id	integer	ID користувача	Не може бути порожнім. Внутрішній ключ
email	string	Email користувача	Не може бути порожнім
password	string	Пароль користувача	Не може бути порожнім
created_at	datetime	Дата створення користувача	Не може бути порожнім
updated_at	datetime	Дата останнього оновлення запису користувача	
current_sign_in_at	datetime	Дата поточного входу в систему	
last_sign_in_at	datetime	Дата останнього входу в систему	
current_sign_in_ip	string	IP-адреса поточного входу в систему	

Продовження Таблиці 1.6

1	2	3	4
last_sign_in_ ip	string	IP-адреса останнього входу в систему	
reset_ password_ token	string	Ключ для відновлення паролю	
reset_ password_ sent_at	datetime	Дата останнього відновлення паролю	
sign_in_ count	integer	Кількість входів в систему	

В таблиці requests зберігається інформація про створені заявки у системі та їх статуси (таблиця 1.7).

Таблиця 1.7 – Поля таблиці requests

Параметр	**Тип поля**	**Короткий опис**	**Обмеження**
1	**2**	**3**	**4**
id	integer	ID запиту на аналіз	Не може бути порожнім. Внутрішній ключ

Продовження Таблиці 1.7

1	2	3	4
created_at	datetime	Дата створення запису	Не може бути порожнім
updated_at	datetime	Дата останнього оновлення запису	Не може бути порожнім
user_id	integer	Id користувача створившого запит	Не може бути порожнім. Зовнішній ключ.
status	string	Статус обробки запиту	Не може бути порожнім
request_goal	text	Мета з якою користувач хоче виконати аналіз	Не може бути порожнім

В таблиці request_results зберігається інформація про виконані заявки, містяться посилання на файли з результатами аналізу (таблиця 1.8).

Таблиця 1.8 – Поля таблиці request_results

Параметр	**Тип поля**	**Короткий опис**	**Обмеження**
id	integer	ID	Не може бути порожнім. Внутрішній ключ
created_at	datetime	Дата створення запису	Не може бути порожнім
updated_at	datetime	Дата останнього оновлення запису	Не може бути порожнім
filename	string	Посилання на файл з результатами аналізу на сервері	Не може бути порожнім
request_id	integer	Id заявки	Не може бути порожнім. Зовнішній ключ.

В таблиці request_state_histories міститься інформація про зміни статусів виконання заявок (таблиця 1.9).

Таблиця 1.9 – Поля таблиці request_state_histories

Параметр	**Тип поля**	**Короткий опис**	**Обмеження**
id	integer	ID	Не може бути порожнім. Внутрішній ключ
from_state	string	Статус з якого було переведено заявку	Не може бути порожнім
to_state	string	Статус в який було переведено заявку	Не може бути порожнім
admin_id	integer	Id адміністратора змінившого статус запиту	Не може бути порожнім. Зовнішній ключ.
created_at	datetime	Дата створення запису	Не може бути порожнім
request_id	integer	Id заявки статус якої було змінено	Не може бути порожнім. Зовнішній ключ.

В таблиці twitter_users зберігається інформація про користувачів мережі Twitter, які були проаналізовані (таблиця 1.10).

Таблиця 1.10 – Поля таблиці twitter_users

Параметр	**Тип поля**	**Короткий опис**	**Обмеження**
id	integer	ID користувача Twitter	Не може бути порожнім. Внутрішній ключ
username	string	Ім’я користувача в мережі Twitter	Не може бути порожнім
tweets_ agregated	text	Агрегований набір твітів користувача	
tweets_ keywords	string	Слова які відображають основну тематику блогу користувача Twitter	
request_id	integer	Id заявки в рамках аналізу якої було проаналізовано користувача	Не може бути порожнім. Зовнішній ключ.

Висновки до розділу

В розділі описано основні бізнес процес, сформульовано постановка задачі, наведені рішення, які застосовуються з інформаційного забезпечення.

В описі бізнес процесів наведено причини чому обрано Twitter. Наведено як саме відбувається взаємодія між користувачами всередині цієї соціальної мережі.

Описано основні процеси діяльності, передбачені системою, наведені діаграми діяльності.

Наведено актори та їх способи взаємодії з системою.

2 МОДЕЛІ ТА МЕТОДИ КЛАСТЕРИЗАЦІЇ КОРИСТУВАЧІВ СОЦІАЛЬНИХ МЕРЕЖ

2.1 Змістовна постановка задачі

Власна статистика мережі Twitter показує, що на сьогоднішній день активними є 284 мільйони користувачів щороку, і щодня надсилається близько 500 мільйонів твітів [3]. Twitter - це соціальна мережа мікро-блогів, за допомогою якої люди можуть спілкуватися та ділитися своїми думками, використовуючи повідомлення, також відомі як твіти. Twitter є платформою для спілкування не тільки звичайних користувачів, але і для знаменитостей та спілок людей. Таким чином, Twitter є також і потужним маркетинговим інструментом, який використовують організації для створення іміджу брендів та отримання відгуків від клієнтів.

Отже, багато організацій інвестують ресурси в аналіз Twitter з наступних причин:

– рекламувати свої продукти та послуги для залучення більшої кількості клієнтів;

– як дешеве джерело реклами та просування;

– отримати зворотній зв’язок від своїх користувачів;

– щоб зрозуміти свою позицію на ринку.

Аналіз даних твітів та групування їх у важливі категорії складне завдання, оскільки твіти:

– мають обмежену кількість слів через обмежену кількість символів, дозволених Twitter;

– мова їх написання зазвичай неформальна;

– часто твіти включають у собі посилання на зовнішні ресурси.

2.2 Математична модель

Основною задачею є розбиття користувачів Twitter на групи за їх інтересами. Сформулюємо математичну постановку задачі.

Дано:

- множина користувачів *U*;
- *n* – кількість користувачів;
- *k* – кількість кластерів;
- користувач - u_i, де $i = \overline{1,n}$, нехай $u_i \epsilon U$;
- користувачі центри кластерів - u_j, де $j = \overline{1,k}$;
- w_i – вага *i*-того терму у векторній моделі документу.

Змінні:

doc_{u_i}- векторна модель текстової складової користувача;

doc_{u_j}- векторна модель текстової складової користувача, який є центром кластеру;

$sim(doc_{u_i}, doc_{u_j})$ – функція схожості між користувачем та центром кластеру.

Цільова функція для процесу кластеризації є максимізація схожості між користувачами у одному кластері:

$$z = sim\left(doc_{u_i}, doc_{u_j}\right) \rightarrow max, \tag{2.1}$$

де функція схожості визначається таким чином:

$$sim\left(doc_{u_i}, doc_{u_j}\right) = \frac{doc_u doc_{u'}}{|doc_u||doc_{u'}|}. \tag{2.2}$$

Векторна модель текстової складової користувача представлена у такому вигляді:

$$doc_{u_i} = (w_1, w_2 \dots, w_i). \tag{2.3}$$

Обмеження:

$$0 < sim(doc_u, doc_{u'}) < 1. \quad (2.4)$$

Для розв’язку даної задачі обрано метод кластеризації k-means, а для обчислення схожості текстової складової, обрано метод машинного навчання порівняння векторних моделей документів.

2.3 Огляд методів розв’язання

Основною складністю при кластеризації користувачів є вибір методів, за якими буде порівнюватись їхня схожість.

Провівши аналіз існуючих способів кластеризації користувачів було вирішено використовувати схожість користувачів між собою, як оцінку відстані між ними у кластері. Для спрощення роботи та збільшення точності обробки, вирішено брати для аналізу агреговану колекцію всіх твітів користувача.

Індекс схожості визначає відстань між двома текстовими складовими користувачів. Кластерні алгоритми використовують деяку функцію вимірювання подібності як критерій віднесення елементу до певного кластеру. Ці функції допомагають визначити, наскільки схожі або несхожі два користувача.

Перш за все, документи повинні бути перетворені у векторну форму, оскільки алгоритми кластеризації та методи вимірювання дистанції не можуть інтерпретувати документи в їх оригінальній формі. Векторна модель (VSM) - широко використовуваний метод представлення тексту документа. Вага елементу (токену) вектору визначає свій внесок у семантику документа [4].

2.3.1 Попередня обробка повідомленнь

У будь-якому окремому твіті можуть бути терміни або слова, які не є важливими для розгляду при кластеризації, і тому слід видалити непотрібну інформацію, щоб зберегти лише важливі ознаки. Твіттер містить у собі переважно короткі повідомлення, і користувачі, як правило, використовують не

літературну мову. Тому, попередня обробка повідомлень перед аналізом може стати складним завданням. Проте, вона є важливим кроком, оскільки це може вплинути на результати процесу аналізу схожості.

2.3.1.1 Токенізація

Токенізація – це процес розбиття строки або документу на невеликі логічні частини, які називаються токенами.

Наприклад повідомлення такого типу: “@AliBunkall : When Obama took office, 180000 US troops were in global conflict zones”, буде перетворено на такий масив токенів:

[‘@AliBunkall’, ‘:’, ‘When’, ‘Obama’, ‘took’, ‘office’, ‘,’, ‘180000’, ‘US’, ‘troops’, ‘were’, ‘in’, ‘global’, ‘conflict’, ‘zones’] ,

тобто першим етапом твіти будуть розбиті на масиви токенів.

2.3.1.2 Видалення пунктуації

Пунктуація також може створювати зайвий шум, при тому що не несе у собі ніякого смислового навантаження, яке б можна було використати, отже на цьому етапі масив перетворюється у такий:

[‘@AliBunkall’, ‘When’, ‘Obama’, ‘took’, ‘office’, ‘180000’, ‘US’, ‘troops’, ‘were’, ‘in’, ‘global’, ‘conflict’, ‘zones’].

2.3.1.3 Видалення стоп слів

Стоп слова також можуть створювати дуже великий шум, адже вони є дуже часто повторюваними, не маючи у собі ніякої тематики, тобто вони загальні.

Прикладом таких слів для англійської мови буде:

“ 'i', 'me', 'my', 'myself', 'we', 'our', 'he', 'ours', 'ourselves', 'you', 'your', 'yours', 'yourself', 'yourselves', 'him', 'herself', 'its', 'they', 'them', 'themselves', 'what', 'which', 'who', 'whom', 'this', 'that', 'these', 'those', 'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have', 'has', 'had', 'having', 'do', 'does', 'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'if', 'or', 'because', 'as', 'until', 'while', 'of', 'at', 'by', 'for', 'with', 'about', 'against', 'between', 'into', 'through', 'during', 'before', 'after', 'above', 'below', 'to', 'from', 'up',

'down', 'in', 'out', 'on', 'off', 'over', 'under', 'again', 'further', 'then', 'itself', 'once', 'here', 'his', 'there', 'when', 'where', 'why', 'how', 'all', 'any', 'both', 'she', 'each', 'few', 'more', 'most', 'other', 'some', 'such', 'no', 'nor', 'not', 'only', 'own', 'it' 'same', 'so', 'than', 'too', 'very', 's', 't', 'can', 'will', 'just', 'don', 'hers' 'should', 'now'" [5].

Отже після цього етапу масив токенів буде мати такий вигляд:

['@AliBunkall', 'Obama', 'took', 'office', '180000', 'US', 'troops', 'global', 'conflict', 'zones'].

2.3.1.4 Видалення шуму з повідомлень

Після виконання попередніх кроків, все одно можуть залишатись деякі неважливі слова. Наприклад, слова з довжиною менше трьох символів. У деяких випадках короткі слова також можуть бути актуальними. Однак у цьому сценарії функції короткої довжини не є релевантними для процесу знаходження тематики і їх слід видалити. Таким чином, для видалення слів довжиною менше трьох символів потрібна подальша фільтрація. Іноді алфавітно-цифрові слова, такі як "abc123", "180000", також зустрічаються в документах чи твітах, які не є важливими для класифікації тексту. Ці слова також відкидаються.

Отже на цьому кроці масив токенів матиме такий вигляд:

['@AliBunkall', 'Obama', 'took', 'office', 'troops', 'global', 'conflict', 'zones'].

2.3.1.5 Видалення посилань

У Twitter посилання на зовнішні ресурси або ж на інших користувачів є дуже часто вживаними. Але для знаходження тематики вони будуть лише заважати. Зазвичай ім'я користувача починається з "@" символу а посилання починаються з "http".

Отже, провівши останній етап підготовки твітів отримується масив токенів, кожен із яких може мати якусь вагу у всьому документі та представляти певну тематику:

['Obama', 'took', 'office', 'troops', 'global', 'conflict', 'zones'].

2.3.2 Методи кластеризації

Методи кластеризації ітераційно відокремлюють дані в різні групи або кластери шляхом мінімізації деякої цільової функції. Вони також відомі як методи кластеризації на основі центроїдів. Для того, щоб розділити набір даних, об’єкти порівнюються з центрами кластерів, таким чином, щоб цільова функція була мінімальною або максимальною.

2.3.2.1 Метод k-means

Метод k-means є одним з найпопулярніших алгоритмів кластеризації. Алгоритм складається з наступних кроків [6-8].

ПОКИ кластерні центри не стануть стійкими (не змінюватимуться).

КРОК 1. Обирається кількість кластерів на які буде розбито вхідні дані.

КРОК 2. Випадковим чином обираються центри майбутніх кластерів.

КРОК 3. Кожен елемент приписується до кластеру, відстань до центру якого найменша.

КРОК 4. Розраховується новий центр кластеру як елемент, ознаки якого є середньо-арифметичними серед всіх елементів кластеру.

КІНЕЦЬ ПОКИ.

Мінімізація відстані між елементами кластеру та його центром є цільовою функцією виконання кластеризації:

$$\sum_{i=1}^{n} \min \left|\left|x_j - \mu_i\right|\right|^2, \tag{2.5}$$

де n – це кількість всіх елементів що кластеризуються, x_j – центр кластеру, μ_i – елемент кластеру.

$$j = \overline{1,k}, \tag{2.6}$$

де k – кількість кластерів.

2.3.2.2 Бісективний метод k-means

Інший популярний варіант алгоритму k-means - це бісективний алгоритм k-means. Він отримує результати наступним чином.

ПОКИ необхідна кількість кластерів не досягнута.

КРОК 1. Обирається кластер для розбиття.

КРОК 2. Запускається загальна версія алгоритму k-means, щоб розбити кластер на два підкластери. Цей крок називається бісектуванням.

КРОК 3. Повторювати крок 2 доки не буде досягнена максимальна подібність елементів у кластері.

КІНЕЦЬ ПОКИ.

У результаті [9] зроблено висновок, що бісективний алгоритм розподілу k-means перевершує традиційний k-means з точки зору точності та ефективності. Також вказано, що алгоритм k-means та бісективний алгоритм k-means працює краще, ніж агломерна ієрархічна кластеризація. Висвітлено один великий недолік агломераційного ієрархічного кластеризації. Стверджується, що помилки можуть відбутися на попередніх етапах, і ці помилки дуже погано впливають на весь подальший процес [9].

2.3.2.3 Метод k-medoids

Для вирішення проблеми шуму було розроблено алгоритм k-medoids. Він схожий на алгоритм k-means, але він не приймає середнє значення для всіх елементів у кластері, щоб знайти центроїд. Алгоритм k-mediod вибирає один з елементів у вигляді точки для порівняння [10]. Це вважається обчислювальним дорогим і також неефективним для великих наборів даних [11-12].

2.3.2.4 Метод Mini-Batch k-means

Час обчислення стандартного алгоритму k-means збільшується з збільшенням кількості елементів у великих наборах даних. Таким чином, у роботі [13] запропоновано варіант Mini-Batch k-means, який виконує менше обчислень для великих наборів даних, ніж традиційний алгоритм k-means. Він

використовує міні-партії для зменшення часу, але використовує ту ж цільову функцію, що й алгоритм k-means. Mini-Batch - це невеликі випадкові вибірки з вхідних даних, обраних під час кожної ітерації, що значно скорочує час обчислення.

Основні кроки алгоритму.

ПОКИ існують елементи що не відносяться до кластеру.

КРОК 1. Береться частина вхідних даних, яка разом формує Mini-Batch.

КРОК 2. Mini-Batch призначається найближчому центру кластеру, далі знаходиться новий центр кластеру.

КІНЕЦЬ ПОКИ.

На рис 2.1 показана швидкість обробки з k = 3 і k = 10 у порівнянні традиційного алгоритму k-means з алгоритмом Mini-Batch k-means. Результати показують, що алгоритм Mini-Batch k-means виконується швидше і дає кращі результати навіть на великих наборах даних (рисунок 2.1). І навпаки, традиційний алгоритм k-means виконується повільніше на великих наборах даних.

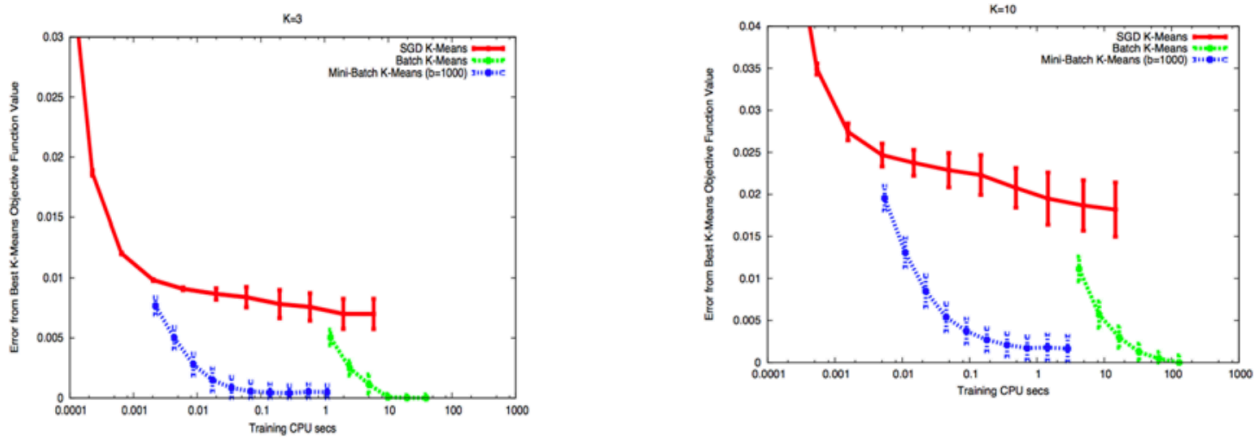

Рисунок 2.1 – Порівняння швидкодії алгоритмів [13]

2.3.3 Методи вимірювання схожості тексту документа

Для обрахування схожості між документами зручно їх звести до векторної моделі, а далі шукати індекс схожості порівнюючи ці вектори. Нижче наведено способи порівняння векторних моделей.

2.3.3.1 Евклідова відстань

Евклідова відстань рахується як сума квадрату різниці між координатами двух об’єктів [15-16]. Тому, Евклідова відстань d між двома n-розмірними векторами X_i та X_j рахується як:

$$d = \sqrt{\sum_{k=1}^{n} (x_{ik} - x_{jk})^2}, \tag{2.7}$$

де x_{ik} – k-тий елемент вектору X_i, а x_{jk} – k-тий елемент вектору X_j.

2.3.3.2 Косинус подібності

Косинус подібності вимірюється як косинус кута між двома векторними моделями документів [15-16]:

$$\cos\theta = \frac{X_i \cdot X_j}{|X_i| \cdot |X_j|} \tag{2.8}$$

2.3.3.3 Коефіцієнт Жаккарда

Коефіцієнт Жаккарда вимірює відстань як перетин, поділений на об'єднання об'єктів. Він обчислює суму ваги загальних термів і порівнює його з сумою ваг термів, що зустрічаються в будь-якому з двох документів, але не є загальними [17-18].

$$jaccard(x, y) = |X_i \cap X_j| / |X_i \cup X_j| \tag{2.9}$$

2.4 Модифікація методу розв’язання задачі

Головною особливістю роботи є те, що змінюється оцінка відстані між елементами кластеру, і використовується замість неї певний індекс схожості. Тобто кластеризація перетворюється у нечітку.

При кластеризації буде порівнюватись не відстань між об’єктами кластеру, а те, наскільки ці об’єкти схожі між собою.

Одним з мінусів є те, що на обрахування цього індексу схожості буде витрачатись багато часу. Це пов’язано з тим, що на цьому етапі застосовуються методи симантичного аналізу, які потребують багато обчислювальних ресурсів.

За рахунок того, що досліджується тематика твітів користувача і схожість між тематиками береться, як оцінка кластеризації, можна сказати що алгоритм є достатньо точним [17-18].

2.5 Розробка алгоритму кластеризації користувачів соціальної мережі

Далі наведено модифікований алгоритм k-means, задачою якого є розбиття користувачів соціальної мережі на кластери, в залежності від схожості документів їх агрегованих твітів.

Крок 1. ВИЗНАЧИТИ початкову кількість кластерів.

Крок 2. ОБРАТИ випадковим чином k центроїдів кластерів.

Крок 3. ПІДГОТУВАТИ документи до аналізу.

Крок 4. ОБЧИСЛИТИ векторну модель документу.

Крок 5. ОБРАХУВАТИ матрицю схожості документів, на основі векторної моделі.

Крок 6. ПРИЗНАЧИТИ кожен документ кластеру, центральний документ якого є найбільш схожим.

Крок 7. ОБЧИСЛИТИ новий центр для кожного кластеру, обравши документ в кластері найбільш схожий на інші.

Крок 8. ЯКЩО кластерні центри стійкі перейти до (9), ІНАКШЕ до (6).

Крок 9. ЗБЕРЕГТИ центри кластерів та матрицю схожості у ньому.

На цьому робота алгоритму завершується.

2.6 Результати досліджень ефективності методу

Ниже наведемо способи порівняння ефективності алгоритмів кластеризації.

Однорідність вимагає, щоб всі кластеризуємі об’єкти були однієї природи. Однорідність та повнота можуть бути обчисленими використовуючи функції ентропії та умовної ентропії *H*. Для її обчислення використовується формула [19]:

$$h = 1 - \frac{H(C|K)}{H(C)}, \tag{2.10}$$

де К – результат кластеризації, С – відомий розподіл на кластери.

Повнота є відношенням числа знайдених об'єктів до всіх об'єктів в базі. Для її обчислення використовується формула [19]:

$$c = 1 - \frac{H(K|C)}{H(K)} \tag{2.11}$$

Для того щоб поєднати в одну оцінку однорідність і повноту використовується V-міра [19]:

$$v = 2\frac{hc}{h + c} \tag{2.12}$$

Також, для оцінки ефективності враховується силует. Значення силуету - це показник того, наскільки об'єкт схожий з власним кластером (згуртованість) у порівнянні зі схожістю з іншими кластерами (розділення). Для визначення силуету непотрібний відомий розподіл на кластери [19]:

$$s = \frac{b - a}{\max(a, b)}, \tag{2.13}$$

де a – середня відстань від кожного об’єкта до інших об’єктів в межах одного кластеру, а b – середня відстань від кожного об’єкта до об’єктів найближчого кластеру.

Можна побачити, що модифікований метод дає гірші результати по швидкодії, але кращі по точності (таблиця 2.1). Для даної роботи це не є проблемою, адже сфера застосування передбачає, що користувач готовий очікувати деякий час.

Таблиця 2.1 – Порівняння методів кластеризації

	Гомогенність	**Повнота**	**V-міра**	**Силует**	**Час виконаня**
k-means	0.723337	0.732861	0.727180	0.182158	22,32
Бісективний k-means	0.814232	0.836367	0.825151	0.181289	20,21
k-medoid	0.907307	0.766901	0.821467	0.112199	38,59
Власний модифікований метод k-means	0.950012	0.781301	0.864857	0.116385	39,33

Висновки до розділу

У цьому розділі наведено моделі та методи, які використовуються у роботі. Наведено формалізований опис задачі.

Наведено опис, як підготувати дані для аналізу для зменшення шумів при аналізі. Наведено опис методів кластеризації. Наведено опис методів вимірювання схожості документів між собою.

Описано спосіб модифікації існуючого алгоритму. Побудовано алгоритм розв’язання. Досліджено ефективність методів, порівняно їх ефективність з власною модифікацією.

Зроблено висновок, що швидкодія модифікованого алгоритму гірша ніж у інших, але точність виконання кластеризації забезпечується тим, що відстань від центру кластеру до елементу рахується як оцінка схожості цих елементів.

3 ОПИС ПРОГРАМНОГО ТА ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Засоби розробки

При розробці даного програмного продукту були використані засоби для програмування бекенд частини: Ruby та фрейморк Rails, а також база даних PostgreSQL. Також буде використано Twitter API.

Рубі – це мова створена Матцом з метою створити мову більш потужну, ніж Perl, і більш об'єктно-орієнтовану ніж Python [20]. Ruby використовується в ряді високопрофесійних додатків, зокрема: моделювання в дослідницькому центрі НАСА Ленглі, моделювання для дослідницької групи Motorola, як API для мікро-скриптів для Google SketchUp, і як єдина мова програмування, використовується для розробки веб-сайту, для управління проектами Basecamp. Ruby – це об'єктно-орієнтована мова програмування. В Ruby будь-яке значення, включаючи числові літерали, а також значення true і false, є об'єктом. Щоб отримати доступ до внутрішнього стану об'єкта потрібно використовувати метод доступу. Дужки, які зазвичай знаходяться в методах та функціях інших мов програмування, тут не потрібні, особливо якщо аргументи не потрібні. Щоб прискорити розробку додатків, Ruby може використовуватися разом із IDE (Integrated Development Environment). Це дозволить програмісту з відносною легкістю писати, запускати та налагоджувати програми. Ruby можна запустити на Windows, Linux, Mac або Solaris. Ruby-програми та бібліотеки, як правило, випускаються як файли gem. Переважно поширюються за допомогою системи упаковки RubyGems. Вихідний код мови Ruby може вільно завантажуватись, використовуватись, копіюватись, модифікуватись та розповсюджуватись.

Ці та інші особливості Ruby роблять розгортання додатків надзвичайно швидким

Rails - бібліотека програмного забезпечення, яка розширює мову програмування Ruby.

Rails – це програмний код, який додається до мови програмування Ruby у вигляді бібліотеки пакетів (зокрема, RubyGem), яка встановлюється за допомогою інтерфейсу командного рядка операційної системи.

Rails є основою для створення веб-сайтів. Rails встановлює конвенції для полегшення співпраці та підтримки. Ці конвенції кодуються як API Rails (інтерфейс прикладного програмування або директиви, що керують кодом). API Rails задокументовано в Інтернеті та описано в книгах, статтях та публікаціях в блозі [21].

Rails об'єднує мову програмування Ruby з HTML, CSS та JavaScript, для створення веб-додатків. Оскільки Rails працює на веб-сервері, він вважається платформою для розробки веб-додатків на серверній стороні.

Rails в більшій мірі - це більше, ніж бібліотека програмного забезпечення та API. Rails - це центральний проект великої спільноти, що створює бібліотеки програмного забезпечення, які спрощують завдання створення складних веб-сайтів. Члени спільноти Rails мають багато основних цінностей, часто використовують ті самі інструменти. Rails користується популярністю серед веб-стартапів, переважно тому, що пул відкритих програмних бібліотек, таких як RubyGems, дає змогу швидко створювати складні сайти.

Для спрощення розробки панелі адміністратора було використано бібліотеку ActiveAdmin [22].

Sidekiq бібліотека, яка надає можливість виконувати деякі дії у фоні. Sidekiq використовує потоки для обробки багатьох завдань одночасно в одному процесі [23].

PostgreSQL - реляційна система керування базами даних. На відміну від більшості інших СУБД, підтримка та розробка PostgreSQL відбувається завдяки співпраці розробників та великих компаній. Саме тому, ця СУБД широко використовується та в ній застосовуються найновіші досягнення. Великою перевагою СУБД PostgeSQL є безліч вбудованих типів даних, її швидкість, та простота у розширенні. Також, вона має високу надійність.

Сервер PostgreSQL розроблено за допомогою імперативної мови програмування C. Для того щоб розвернути власний екземпляр серверу, потрібно завантажити вихідний код серверу (зазвичай поданий у вигляді текстових файлів). Далі файли необхідно скомпілювати та скопіювати в каталог зі всіма програмами. Детальна інструкція наведена в документації [24].

PostgreSQL пропонує можливості індексування, яких немає у інших БД з відкритим вихідним кодом. Крім стандартних індексів, він підтримує часткові індекси, функціональні індекси, GiST і GIN індекси.

Для прикладу в інших базах даних індекси реалізовані значно гірше. В MySQL 5.7.6 були представлені генеруємі стовпці, які можна використовувати як функціональні індекси. У MariaDB віртуальні (також відомі як генеруємі або обчислювані) стовпці з'явилися у версії 5.2, але підтримують тільки використання вбудованих функцій для створення стовпців (визначені користувачем функції відсутні). У версії 2.0 Firebird було представлено індексування виразів за допомогою обчислюваних стовпців. Проте, жодна з цих баз даних не підтримує часткові, GiST або GIN індекси. Крім того, нативні типи даних JSON не можуть бути проіндексовані в цих базах даних.

Матеріалізовані подання (Materialized views) - це ще одна зручна функція віртуальних таблиць, підтримувана PostgreSQL. Вони, як і звичайні подання, представляють результат запиту, який буде часто використовуватися, але різниця в тому, що результат зберігається на диску як звичайна таблиця. Матеріалізовані подання можуть бути проіндексовані. Крім того, на відміну від звичайних подань, які перебудовуються кожен раз коли їх викликають, матеріалізовані подання фіксуються в часі зі збереженим результатом. Вони не оновлюються, якщо не робити це навмисно. Це може істотно збільшити швидкість з якою здійснюються запити, що використовують матеріалізовані подання. Замість використання звичайних подань або необхідності здійснювати складні об'єднання таблиць або використовувати групуючі функції в запиті, можна використовувати матеріалізовані подання, де всі необхідні дані вже підготовлені і чекають на диску. Коли знадобиться оновити дані в

матеріалізованих поданнях, це можна буде зробити на вимогу за допомогою команди REFRESH.

Для реалізації графічного користувацького інтерфейсу були використані HTML, CSS, JavaScript.

HTML – мова, яка використовується для створення розмітки веб-сторінки. Розмітка майже всіх веб-сторінок створена мовою HTML. Після того, як сервер надає браузеру HTML документ, браузер починає обробку цього документу, та у результаті відображає сайт у звичному для людини вигляді [25].

CSS – мова, за допомогою якої стилізується відображення розмітки веб-сторінки. Блочна верстка замінила табличну верстку. Головною перевагою є те, що зміст сторінки, тобто розмітка, і візуальні налаштування є повністю розділеними [26].

JavaScript – це динамічна мова програмування, за допомогою якої HTML документ може набути динамічної інтерактивності. Його винайшов Брендан Ейч, співзасновник проекту Mozilla, Фонд Mozilla та корпорація Mozilla. На сьогодні це одна з самих популярних мов програмування.

Важливою перевагою JavaScript є те, що він дуже універсальний. Програми написані на ньому варіюються від галерей з фотографіями, де їх можна листати, до анімованих 2D та 3D ігор. Також JavaScript вміє за допомогою драйверів працювати з базами даних.

Сам JavaScript досить компактна мова програмування. Є важливим те, що спілкою розробників вже написано велику кількість інструментів та бібліотек на основі ядра мови.

Ці інтрументи включають:

– бібліотеки направлені на взаємодію зі сторонніми організаціями, такими як Facebook та Twitter;

– сторонні бібліотеки, які пришвидшують розробку програмних продуктів, за рахунок деякого реалізованого базового функціоналу.

За допомогою цих інтрументів існує можливість розширювати функціонал та можливості мови JavaScript застосовуючи мінімальні зусилля [27].

Мова JavaScript використовується для:

– реалізації інтерактивності на веб-сторінках;

– створення односторінкових веб-застосунків (ReactJS, AngularJS, Vue.js);

– створення веб-серверів;

– програмування застосунків, що встановлюються у систему;

– створення застосунків для мобільних пристроїв.

3.2 Архітектура програмного забезпечення

При розробці було використоно підхід до проектування ахітектури Domain Driven Design (DDD). DDD фокусується на трьох основних принципах:

– зосередження на основній області та логіці домену;

– базові комплексні конструкції на моделях домену;

– постійна співпраця з експертами домену, щоб покращити модель програми та вирішити будь-які нові проблеми, пов'язані з доменом [28].

Основними термінами, які є корисними при описі та обговоренні практик DDD є:

– контекст – це параметр, в якому з'являється слово або твердження, яке визначає його значення. Висловлювання про модель можна зрозуміти лише в контексті;

– модель – це система абстракцій, яка описує вибрані аспекти домену і може використовуватися для вирішення проблем, пов'язаних із цим доменом;

– загальнодоступна мова – це мова, структурована навколо моделі домену, яка використовується розробником, щоб об'єднати всі його дії з програмним забезпеченням;

– обмежений контекст – це опис кордону (як правило, підсистеми), в рамках якої конкретна модель визначена та застосовується.

Дизайн, в основі якого лежить предметна область, також визначає цілий ряд концепцій високого рівня, які можуть бути використані у взаємозв'язку між собою для створення та модифікації моделей домену.

Дизайн на основі домену також сильно підкреслює все більш популярну практику безперервної інтеграції, яка змушує всю команду розробників використовувати один репозиторій в якому збережений код та додавати свої зміни до нього щодня.

3.3 Діаграма класів

ActiveRecord (AR) – це шаблон проектування, який реалізується для спрощення взаємодії з реляційною базою даних. Вперше був описаний Мартіном Фаулером. Об’єкт інферфейсу, що реалізує ActiveRecord обов’язково містить у собі базові функції CRUD. Також, необхідною умовою є пряма відповідність між атрибутами об’єкта та полями у відповідній йому таблиці бази даних.

Завдяки такій реалізації спрощується взаємодія з базою даних. Можна звертатися до бази даних, застосовуючи методи об’єкту. Отже, можна працювати з більш об’єктно-орієнтованим кодом [29].

Переваги використання ActiveRecord:

– написання коду використовуючи ActiveRecord відбувається швидко, в тому випадку, коли властивості об'єкта прямо співвідносяться з полями в таблиці бази даних;

– збереження відбувається в одному місці, що дозволяє працювати з базою даних більш просто.

Недоліки використання ActiveRecord:

– моделі Active Record порушують принципи SOLID [30]. Зокрема, принцип єдиної відповідальності (S в принципах SOLID). Відповідно до принципу, доменний об'єкт повинен мати тільки одну зону відповідальності, тобто тільки свою бізнес-логіку. Викликаючи його для збереження даних, йому

додається додаткова зона відповідальності, збільшуючи складність об'єкта, що ускладнює його підтримку і тестування;

– реалізація збереження даних тісно пов'язана з бізнес-логікою, а це означає, що якщо пізніше потрібно буде використовувати іншу абстракцію для збереження даних (наприклад зберігання даних в XML-файлі, а не в базі даних), то доведеться робити рефакторинг коду.

Так як в бібліотеці Rails використовується паттерн ActiveRecord, діаграма класів повністю зібагатиметься з ER-діаграмою.

3.4 Діаграма послідовності

У цьому розділі, використовуючи діаграму послідовності, наведено процеси діяльності. які відбуваються у системі.

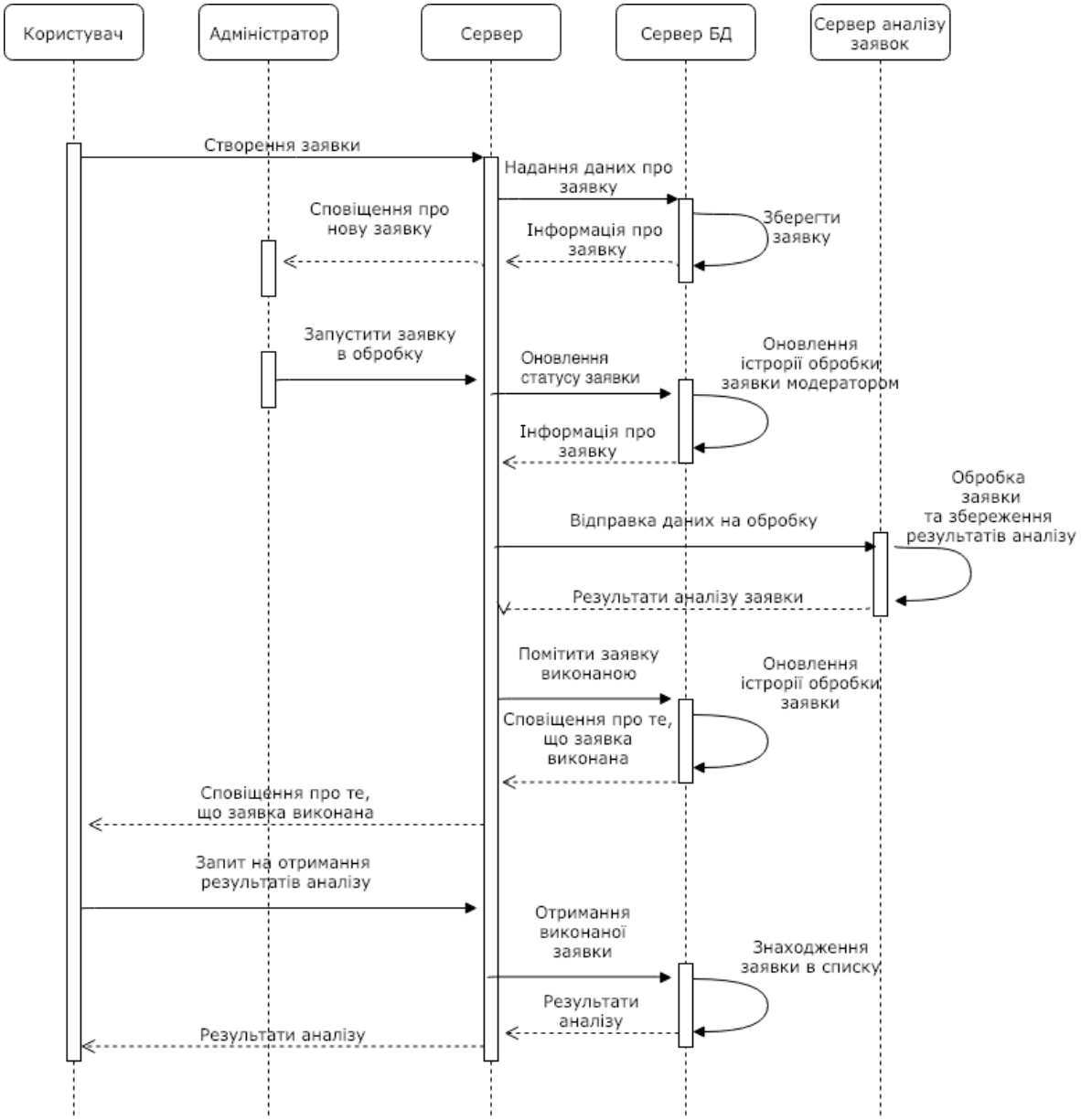

Рисунок 3.1 – Діаграма послідовності

Нижче описано класи, які приймають участь у процесах діяльності та їх відповідальність (таблиця 3.1).

Таблиця 3.1– Перелік класів діаграми послідовності

Клас	**Відповідальність**
1	**2**
Користувач	Створення заявки на аналіз, отримання результатів аналізу

Продовження таблиці 3.1

1	2
Адміністратор	Модерація заявок на аналіз
Сервер	Зв’язуюча система всіх компонентів, відповідає на запити користувачів та адміністраторів. Зберігає у собі логіку по взаємодії з БД та сервісом обробки заявок.
Сервер БД	Сервер БД надає серверу доступ до даних у таблицях. Відповідає за обробку запитів на отримання чи зміну даних.
Сервер обробки заявок	Містить у собі всю логіку по якій заявки обробляються. Відповідає про сповіщення серверу та користувача у випадку, коли заявка оброблена.

3.5 Діаграма компонентів

Основою роботи фрейморку Rails є шаблон MVC.

Моде́ль– представлення –контро́лер – один з архітектурних шаблонів, які використовуються у бібліотеці Ruby on Rails для спрощення розробки та проектування програмного забезпечення.

У цьому шаблоні передбачений розподіл системи на три основні частини: модель (набір атрибутів та методів взаємодії з сутністю), подання (інтерфейс який отримує користувач) та контролеру (реалізує модуль керування, поєднує модель з відображенням) [31].

Застосовується для того, щоб зменшити вплив змін на одному з рівнів на інші.

На діаграмі компонентів зображено складові частини системи та способи їх взаємодії між собою (рисунок 3.2).

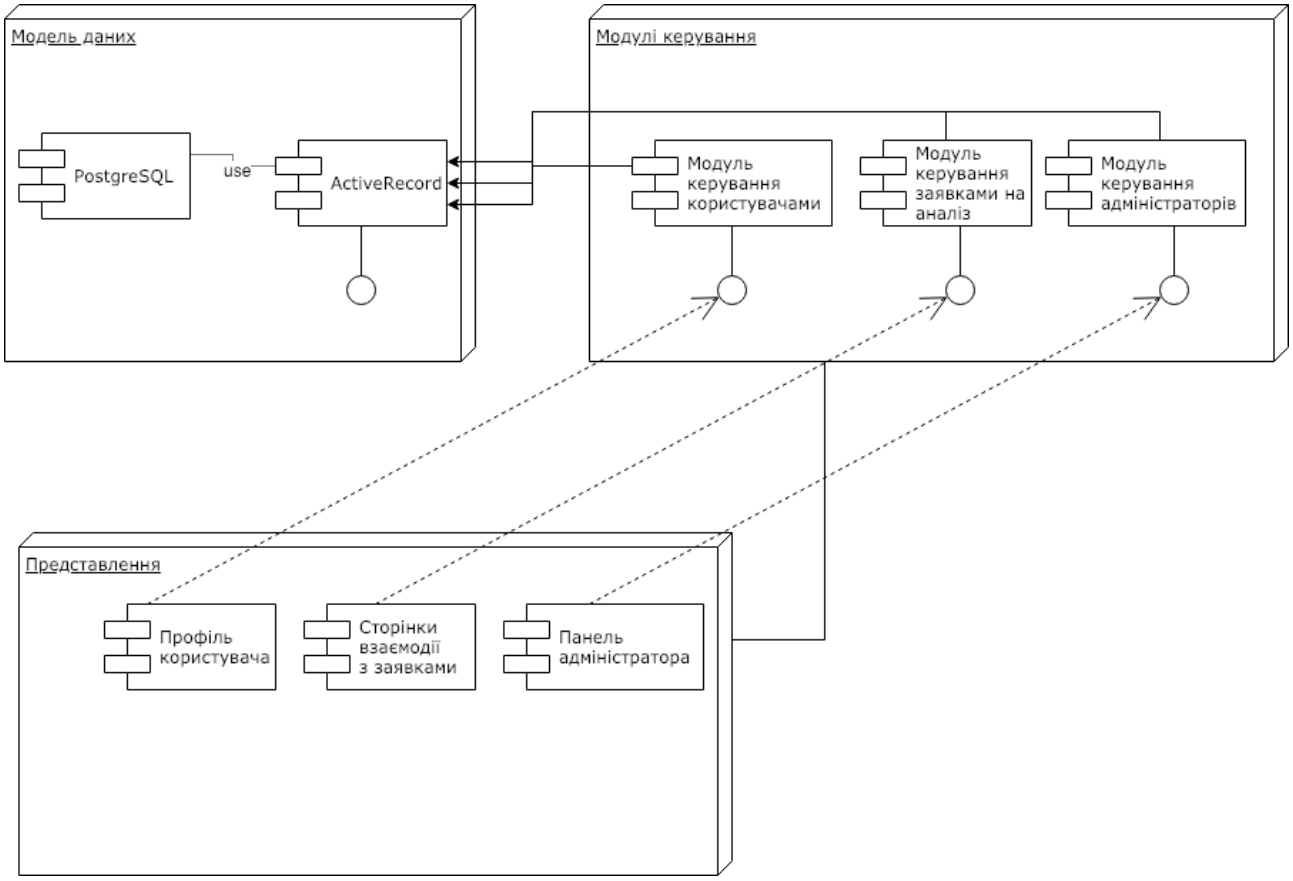

Рисунок 3.2 – Діаграма компонентів

3.6 Інструкція користувача

3.6.1 Реєстрація та авторизація в системі

Мова інтерфейсу система є англійською, так як цільова аудиторія системи англомовна.

Перш за все від користувачів вимагається зареєструватись в системі. Так як режиму гостя не передбачено, при першому вході користувач одразу потрапляє на форму реєстрації (рисунок 3.3).

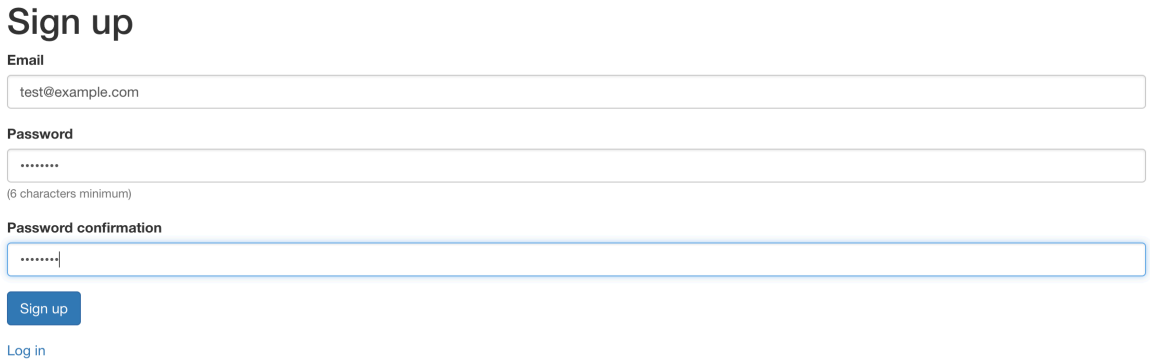

Рисунок 3.3 – Скріншот форми реєстрації

Якщо користувач був зареєстрований раніше, він може перейти на форму авторизації, натиснувши кнопку “Log in”, де він може ввести свої дані (рисунок 3.4) і увійти до свого особистого кабінету.

Рисунок 3.4 – Скріншот форми реєстрації

3.6.2 Створення заявки на аналіз

Після проходження кроків авторизації, користувач потрапляє на головну сторінку профілю (рисунок 3.5).

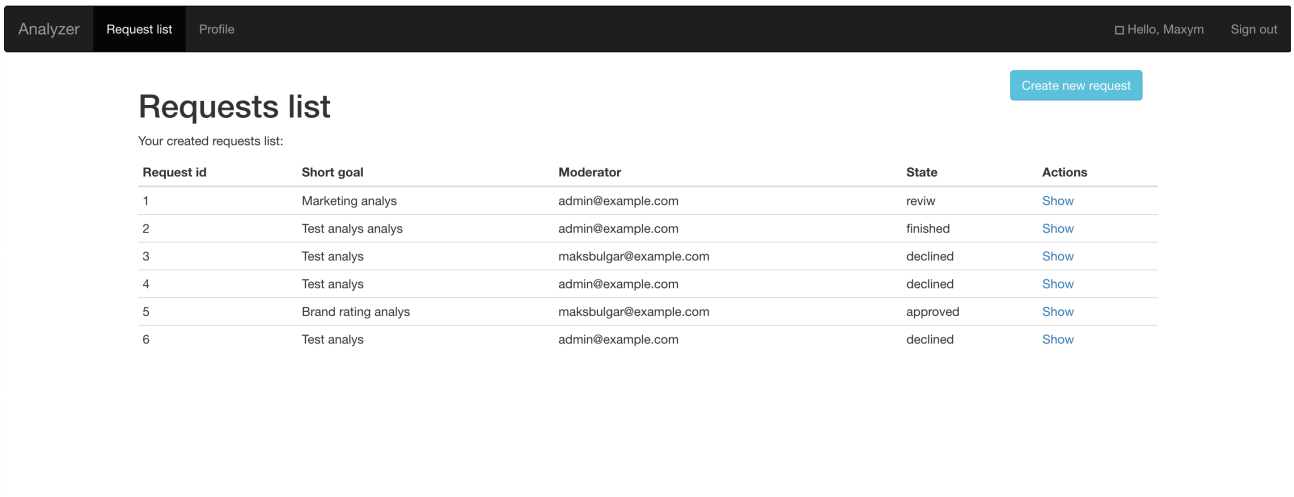

Рисунок 3.5 – Скріншот головної сторінки профілю

На цій сторінці відображено список створених користувачем заявок на аналіз. По кожній заявці в списку відображається така інформація:

– ідентифікатор заявки в системі;

– ціль заявки на аналіз;

– якщо заявка була промодерована, адміністратор, який модерував заявку, інакше поле залишається пустим;

– статус заявки.

При натисканні кнопки “Create new request” користувач перенаправлений на сторінку створення заявки на аналіз (рисунок 3.6).

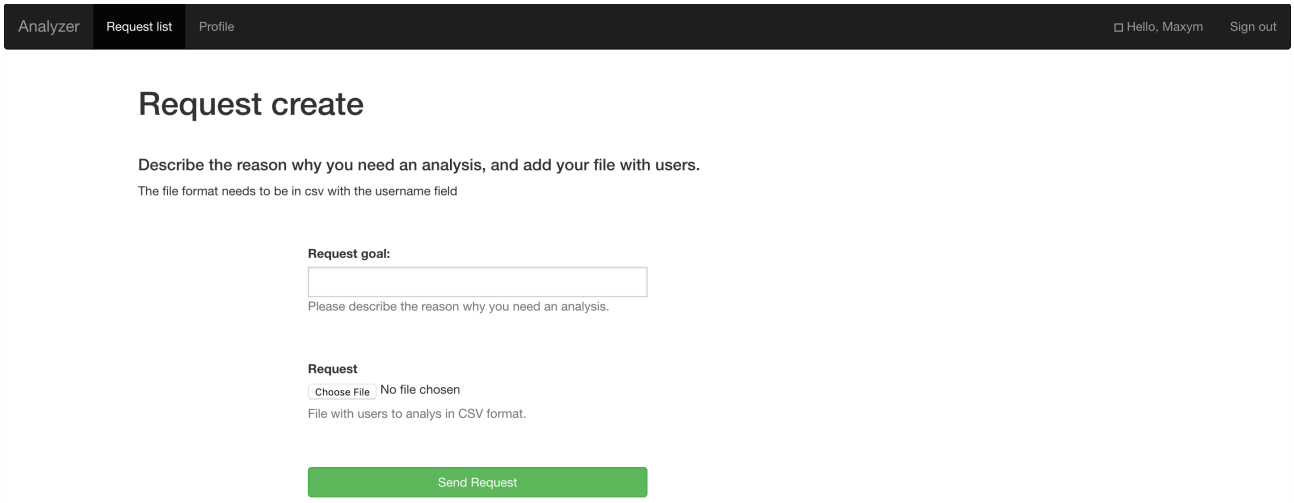

Рисунок 3.6 – Скріншот сторінки створення заявки на аналіз

Користувач вводить опис для чого планується використати аналіз, та прикріпляє файл з вхідними даними (інформація про користувачів, яких потрібно проаналізувати) для аналізу. Після цього він має можливість відправити заявку на модерацію адміністратору натиснувши кнопку “Send Request”.

3.6.3 Отримання результатів аналізу

Після того, як заявка оброблена, користувач отримує на пошту лист. Після цього, користувач з особистого кабінету (рисунок 3.5) може перейти на сторінку перегляду детальної інформації про заявку (рисунок 3.7).

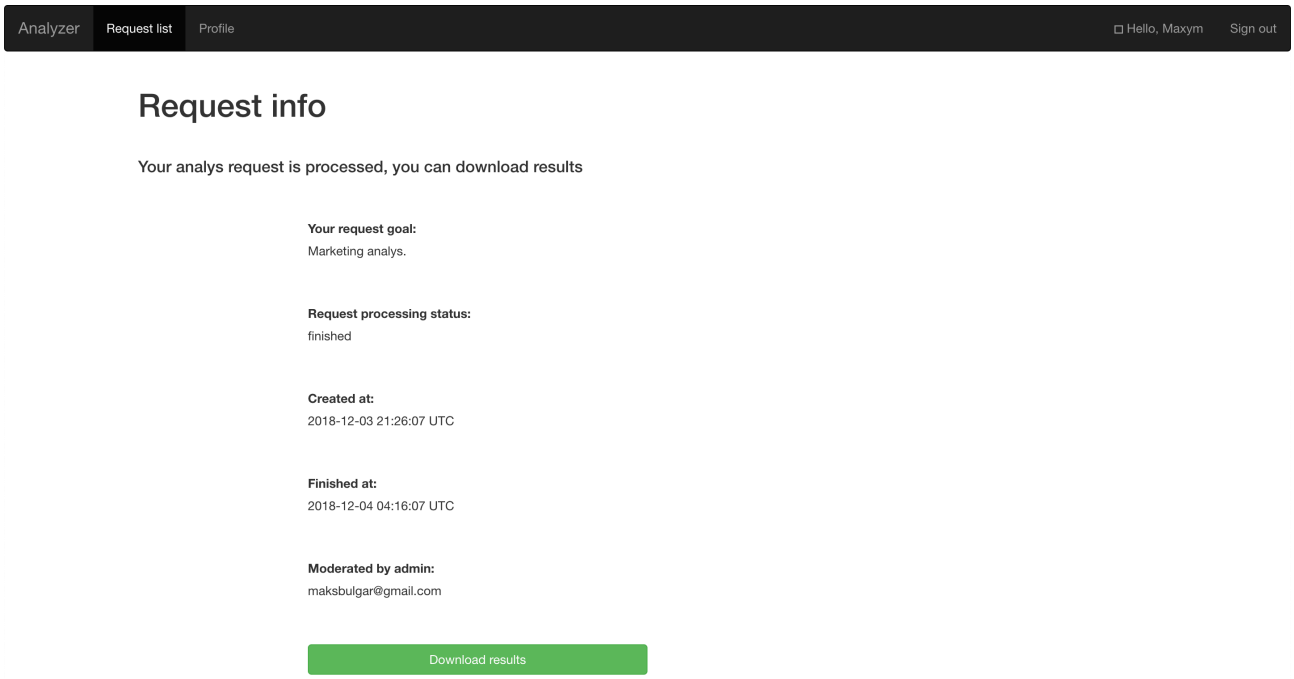

Рисунок 3.7 – Скріншот сторінки перегляду результатів аналізу

Далі натиснувши кнопку “Download results” користувач може завантажити файл з вихідними даними результатів аналізу. Файл з результатами аналізу зберігається форматі CSV, з інформацією про ім’я користувача в мережі Twitter, ключові інтереси цього користувача, та назва групи інтересів, до якої він відноситься.

3.7 Інструкція адміністратора

3.7.1 Особистий кабінет

Після авторизації у адмін панелі, адміністратор потрапляє до особистого кабінету адміністратора (рисунок 3.8).

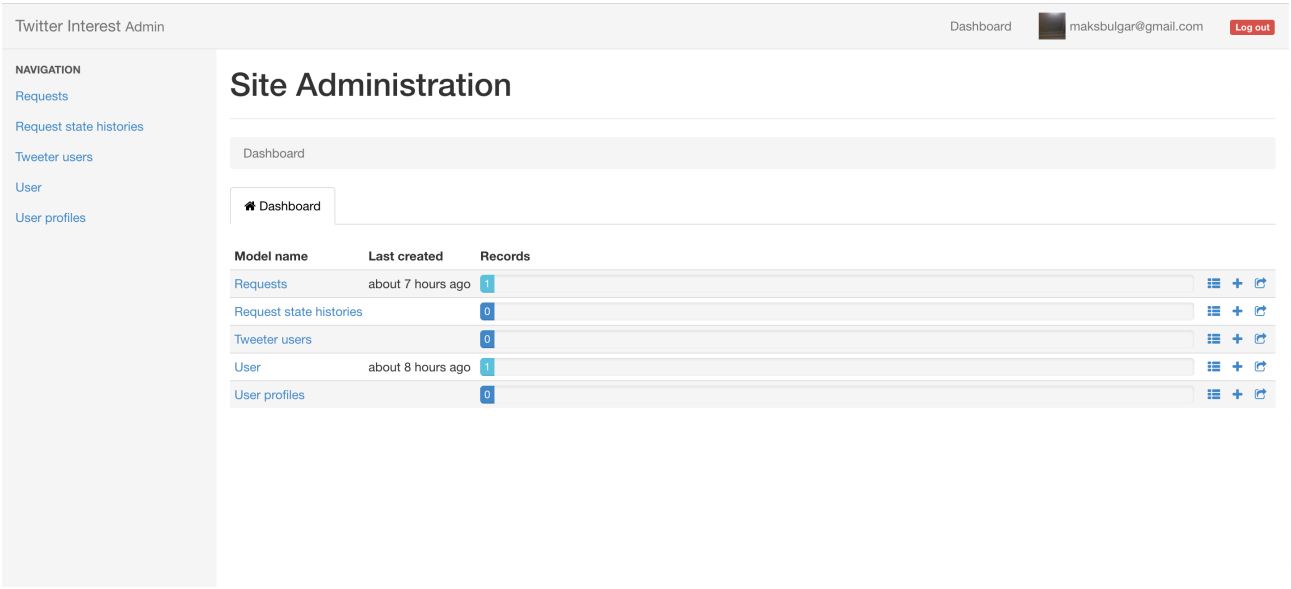

Рисунок 3.8 – Скріншот сторінки особистого кабінету адміністратора

Звідси адміністратор має змогу керувати всіма сутностями у системі.

3.7.2 Модерація заявки

При створені нової заявки адміністратор отримує сповіщення про те, що потрібна модерація. Це сповіщення містить посилання на сторінку редагування інформації по заявці (рисунок 3.9).

На цій сторінці адміністратор бачить всю необхідну для модерації інформацію. Може завантажити файл з даними для аналізу. Змінює статус на бажаний, тобто, або запускає заявку в обробку, або відхиляє її.

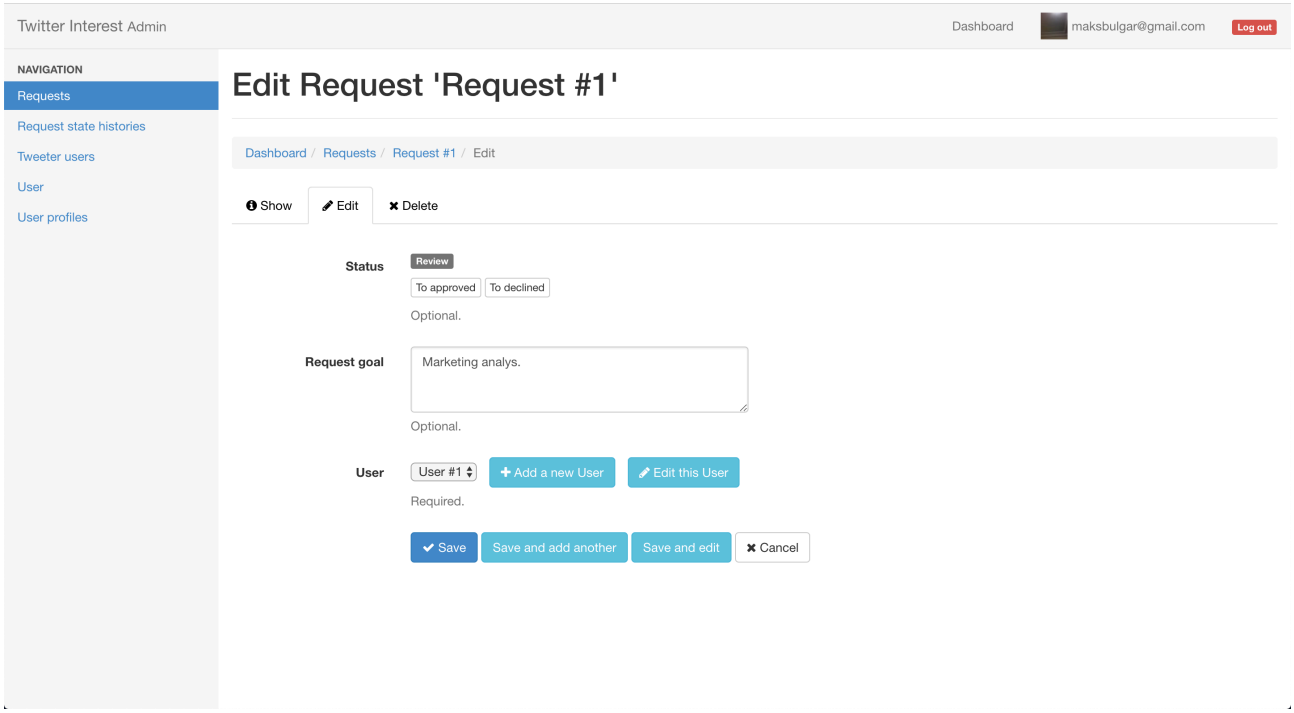

Рисунок 3.9 – Скріншот сторінки модерації заявки

3.8 Опис технічного забезпечення

На діаграмі розгортання зображено те, як сервери частин системи взаємодіють між собою та користувачем (рисунок 3.10).

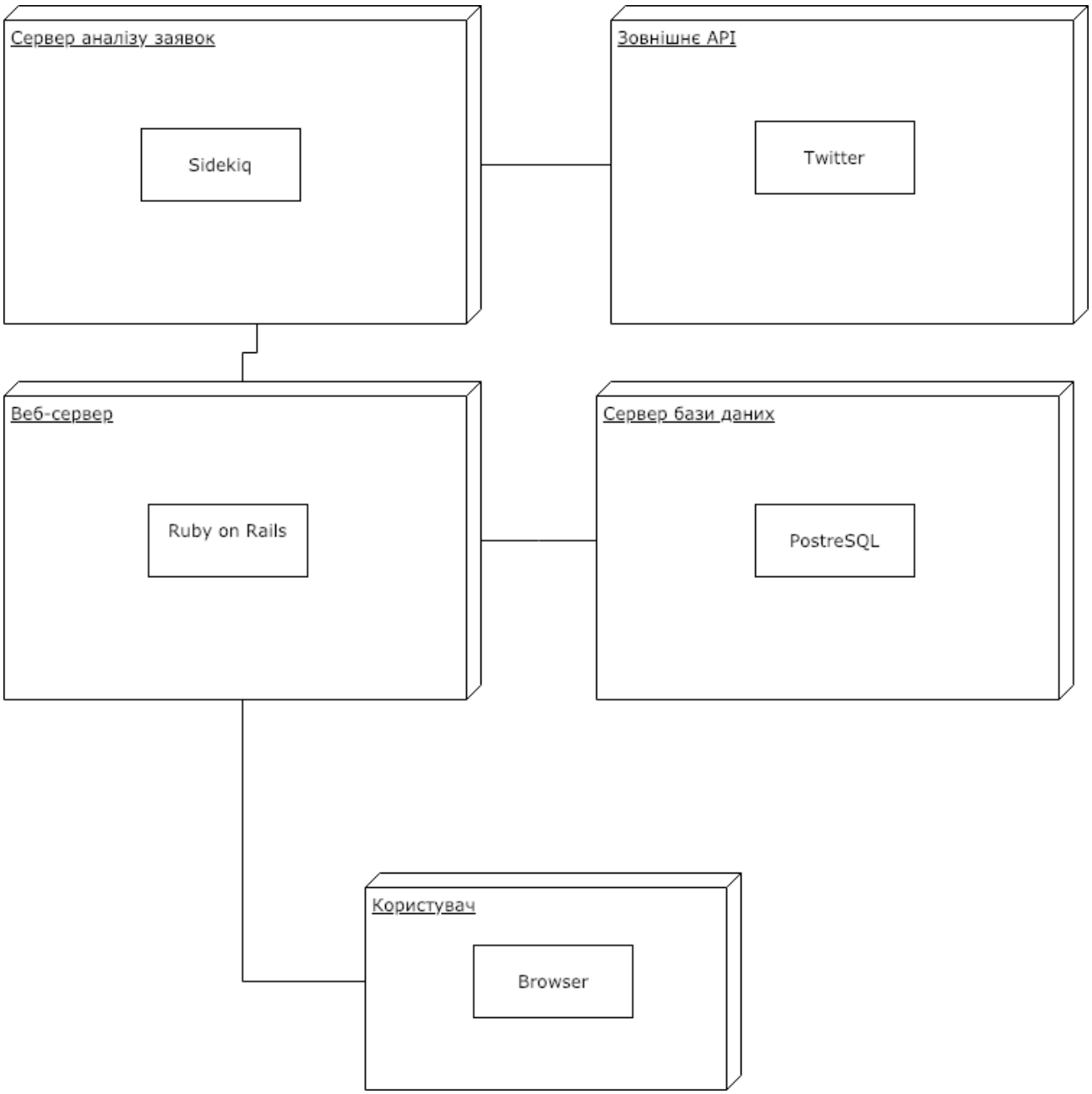

Рисунок 3.10 – Діаграма розгортання

Структура технічних засобів залежить від способу взаємодії користувача з системою, завдань поставлених до системи, вимог до захищеності, можливості інтегрування та ресурсів, доступних до застосування.

Вимоги до технічних засобів складаються з урахуванням можливості виконання встановлених задач системою.

Для правильної та швидкої роботи розробленої системи потрібен сервер, який буде обробляти дані та виступати в ролі API, і який має мати наступні конфігурації:

– процесор тактова частота якого не є нижчою 2,5 ГГц;
– об’єм оперативної пам’яті більше 8 ГБ;
– об’єм HDD або SSD більше 100Гб;
– доступ до мережі Інтернет.

Для коректної роботи клієнтської частини необхідно комп’ютер, що задовольняє таким характеристикам:

– процесор тактова частота якого не є нижчою 1 ГГц;
– об’єм оперативної пам’яті більше 1024 МБ;
– об’єм HDD або SSD більше 20Гб;
– доступ до мережі Інтернет.

Для роботи клієнта мають використовуватись наступні периферійні пристрої комп’ютера:

– монітор;
– мишка;
– клавіатура.

Додатково для роботи клієнта з додатком має бути встановлене таке програмне забезпечення:

– Google Chrome, або будь-який браузер, оновлення якого було здійснене не раніше 2015 року.

Висновки до розділу

В даному розділі наведено засоби розробки. Показано спроектовану в процесі розробки архітектуру. Також, наведено діаграми класів, послідовності та компонентів.

Наведено використані засоби розробки, наведено причини їх використання.

Надано керівництва користувача та адміністратора з рисунками прикладу роботи програми. Описано етапи взаємодії з системою.

Як основний засіб розробки було використано фреймворк Ruby on Rails, так як він задовільняє потреби системи у швидкодії та відмовостійкості.

4 РОЗРОБКА СТАРТАП-ПРОЕКТУ

У даному розділі проводиться аналіз можливості ринкового впровадження продукту. Проаналізовано складнощі, які можуть виникнути з виходом на ринок.

4.1 Опис ідеї проекту

Провівши аналіз ринку, було з’ясовано що прямих конкурентів система не має, так як особливістю систему є те, що в увагу береться переважно аналіз текстової складової, та групування користувачів у групи. Проте враховуючи те, що основою є аналіз даних про користувачів Twitter важливо буде розглянути таких конкурентів (таблиця 4.1):

– TweepsMap – система візуалізації даних про користувачів Twitter на мапі, з врахування безлічі фільтрів;

– Audiense – система аналізу користувачів Twitter, які входять у коло фоловерів користувача системи;

– Twitonomy – система аналізу поведінки користувачів або їх групи.

Таблиця 4.1 – Опис ідеї стартап-проекту

Зміст ідеї	**Напрямки застосування**	**Вигоди для користувача**
1	**2**	**3**
Класифікація користувачів соціальної мережі Twitter, на основі текстової складової їх поведінки.	1. Рекомендація нових інтересів для користувача	Надання користувачу рекомендацій щодо ноових інтересів, які потенціально будуть цікавими для нього.

Продовження Таблиці 4.1

1	2	3
	2. Надання таргетованої для групи користувачів зі схожими інтересами	Надання реклами тим користувачам, які найбільш ймовірно є споживачами даної продукції. Може бути корисно тим, що відсоток отримання прибутку зі збільшенням точності категорії користувачів прямо пропорційно буде збільшуватись.
	3.Отримання реклами яка направлена на інтереси	Отримання найбільш рентабельних рекламних пропозицій для користувача, які можуть стати дійсно корисними та зацікавити його.
	4. Рекомендація схожих користувачів	Надання користувачу рекомендацій за ким йому можливо варто почати спостерігати в соціальній мережі.
	5. Перевірка цільової аудиторії бренду	Отримання інформації про цільову аудиторію бренду. Може бути корисним при плані подальшої місії просування бренду.
	6. Аналіз поведінки груп користувачів	Дослідження трендів серед певної групи користувачів, які мають психологічний вплив на велику аудиторію.

Перелік слабких та сильних характеристик може бути дуже корисним при плануванні наступних етапів у розвитку проекту (таблиця 4.2).

Таблиця 4.2 – Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/п	**Технікоекономічні характеристики ідеї**	**(потенційні)товари/концепції конкурентів**				**W (слабка сторона)**	**N (нейтральна сторона)**	**S (сильна сторона)**
		Мій проект	**TweepsMap**	**Audiense**	**Twitonomy**			
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
1	Вартість обслуговування	Відносно невелика	Не знайдено інформації	Не знайдено інформації	Не знайдено інформації		+	
2	Вартість експлуатації	В місяць 100$	Не знайдено інформації	Не знайдено інформації	Не знайдено інформації		+	

Продовження Таблиці 4.2

1	2	3	4	5	6	7	8	9
3	Безвідмовність	З врахування тогоо що система розподілена, багато процесів виконуються у бекграунді, та сама система написана на широко підтримуємій мові, ймовірність виходу із стробю всієї системи невелика.	Достатня	Досить висока	Достатня		+	
4	Ремонтопридатність	Так як використовуються інструменти, доволі популярні та з великою кількістю розробників, складність ремонту є дуже невеликою. Також є важливим те, що більшість задач виконуються ізольовано та розподілено, тому буде просто знайти де система вийшла зі строю.	Не знайдено інформації про архітектуру.	Не знайдено інформації	Не знайдено інформації		+	

Продовження Таблиці 4.2

1	2	3	4	5	6	7	8	9
5	Оплата праці	Так як дана система являється стартапом, то найбільше ресурсів вимагається на початкових етапах. В подальшому планується пошук інвестицій, та можливо розширення команди для додання нового функціоналу.	Висока	Дуже висока	Висо-ка			+
6	Зручність користування	Так як система у стадії MVP, то зручність користування моментами є не достатньою. В подальшому може знадобитись консультація UX-дизайнеру для покращення зручності користуванням.	Висока	Дуже висока	Висо-ка	+		

Продовження Таблиці 4.2

1	2	3	4	5	6	7	8	9
7	Простота освоєння	Так як для початку було розроблено MVP версію системи, то вона є максимально простою в освоєнні, так як пропонується лише базовий фунціонал.	Висока	Висока	Висо-ка			+
8	Дизайн ресурсу	На перших стадіях не стояв, як вимога необхідна для виходу на ринок. Тому ця частина системи повинна буде бути доопрацьована.	Серед ня	Висока	Висо-ка	+		
9	Відповідність патернами дизайну	Висока	Висока	Висока	Серед ня	+		
10	Відповідність тенденціям дизайну	Середня	Висока	Дуже висока	Серед ня	+		

Продовження Таблиці 4.2

1	2	3	4	5	6	7	8	9
11	Безпека даних користувача	Висока, так, як потрібно виключити можливість, коли користувача взламано, та за його рахунок проведено аналіз	Висока	Висока	Середня		+	
12	Відмово стійкість системи	Так як система була розроблена з врахуванням сучасних стандартів по відмовостійкості, та застосовуються стабільні, широко поширені інтрументи розробки, можна сказати що відмовостійкість системи буде високою.	Дуже висока	Дуже висока	Середня		+	
13	Підтримка оновлень	Присутня	Відсутня	Присутня	Відсутня			

4.2 Технологічний аудит ідеї проекту

Далі наведено технологічний аудит ідеї проекту. В ньому проаналізовано основні технології за допомогою яких буде можливо реалізовано систему.

З таблиці 4.3 можна побачити, що технічна реалізація проекту є досить трудозатратною, але за рахунок того, що використовуються популярні

технології, та тому, що по цим технологіям є багато інформації, проблем з розробкою виникнути не повинно. На початковому етапі сповільнює швидкість розробки Twitter API, так як в ньому присутній ліміт використання.

Таблиця 4.3 – Технологічна здійсненність ідеї проекту

№ п/п	**Ідея проекту**	**Технології її реалізації**	**Наявність технологій**	**Доступність технологій**
1	Отримання інформації про користувачів Twitter	Відкрите API Twitter. Є обмеження на кількість запитів які можуть бути зроблені за годину	Наявна	Доступна в достатньому обсязі
2	Порівняння схожості користувачів за текстовою складовою	Методи та засоби аналізу схожості між тектовими документами	Необхідно розробити	Засоби є доступними
3	Об’єднання користувачів у групи	Методи та засоби кластеризації	Необхідно розробити	Засоби є доступними
4	Інформаційна система зі зручним інтерфейсом	Згадані у попередніх розділах засоби розробки	Необхідно розробити	Використовуються засоби, що є вільними для використання
Технології що використовуються для реалізації проекту: rails, javascript, html, ruby, jquery, posgresql.				

4.3 Аналіз ринкових можливостей запуску стартап-проекту

Ринковий аналіз допомагає ефективніше спланувати подальший напрямок розвитку проекту. В його основу береться поточний стан ринку, а також аналізуються вимоги цільової групи споживачів продукту.

Ринок аналізу даних про користувачів соціальних мереж є досить широким. Проте наша система має значну перевагу в тому, що вона є максимально простою та надає дані у вигляді зручному для будь-якого варіанту використання. Є проблемою обмеження Twitter, так як це збільшує час, який необхідний на виконання аналізу (таблиця 4.4).

Таблиця 4.4 – Попередня характеристика потенційного ринку стартап-проекту

№ п/п	**Показники стану ринку (найменування)**	**Характеристика**
1	**2**	**3**
1	Кількість головних гравців, од	3
2	Загальний обсяг продаж	
3	Динаміка ринку (якісна оцінка)	зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Обмеження які існують: 1.Необхідно запезпечити достатньо надійну систему, щоб захистити дані користувачів. 2.Необхідний високий рівень відмовостійкості системи. 3.Необхідно відповідати кращим практикам UI та UX дизайну. 4.Потрібно вирішити проблему з використання API Twitter, так як обмеження значно збільшують час на отримання результатів.

Продовження Таблиці 4.4

1	2	3
5	Специфічні вимоги до стандартизації та сертифікації	Відсутні
6	Середня норма рентабельності в галузі (або по ринку), %	Відсутня

Було проведено аналіз ринку, проаналізовано основних конкурентів. На основі цього було сформовано хто є аудиторією проекту, та що може зацікавити цих користувачів (таблиця 4.5).

Таблиця 4.5 – Характеристика потенційних клієнтів стартап-проекту

№ п/п	**Потреба, що формує ринок**	**Цільова аудиторія (цільові сегменти ринку)**	**Відмінності у поведінці різних потенційних цільових груп клієнтів**	**Вимоги споживачів до товару**
1	**2**	**3**	**4**	**5**
1	Потреба користувачем у розширенні кола інтересів	Рекомендаційні системи які можуть використовувати дані аналізу системи	Дуже сильно залежить від специфіки рекомендаційної системи	1.Захищеність приватності даних 2.Надання коректних та максимально релевантних рекомендацій

Продовження Таблиці 4.4

1	2	3	4	5
2	Таргетинг по певним групам користувачів	Маркетологи та рекламодавці з різних сфер	Сильно залежить від особливостей сфери діяльності	1. Отримання груп користувачів розподілених на групи за загальними інтересами 2.Висока точність того, що користувач дійсно має належати цій групі, так як в залежності від цього може проводитись аналіз успішності.

Далі проаналізовано основні загрози які можуть перешкоджати успішно вийти на ринок. Фактори відсортовані від найбільш важливого до найменш (таблиця 4.6).

Таблиця 4.6 – Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	**2**	**3**	**4**
1	Конкуренти є монополістами на ринку, вже заявили про себе і мають гарну репутацію	Складно з початковою версією проекту перевершити вже стабільні проекту, які давно пройшли етапи становлення та покращили всі свої слабкі місця. Існує складність в тому щоб піднятися в пошуковому рейтингу.	Добре продумана маркетингова стратегія, дозволяти користувачам безкоштовно скористатись сервісом, SEO-оптимізація. Можливо доцільно буде скористатись рекламою та послугами маркетологів.
2	Недостатня зацікавленість проектом цільовою аудиторією. Недовіра до системи.	Важко пояснити користувачу чому йому необхідна саме ця система, також потрібно запевнити користувача про те що автор системи є добросовісним.	Залучення сторонніх маркетологів. Написання статей про способи використання системи з описом всіх її переваг перед конкурентами. Створення дизайну системи який буде викликати довіру у користувача.

Продовження Таблиці 4.6

1	2	3	4
3	Зміна політики надання даних зі сторони Twitter	Можлива змін політики надання даних зі сторони Twitter, за рахунок цього зменшення способів аналізу. Також за рахунок обмеження в даних можливе зменшення точності аналізу.	Відсутня

Також важливо врахувати і фактори можливості як можна зробити старт системи найбільш благополучним (таблиця 4.7).

Таблиця 4.7 – Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	2	3	4
1	Покращення якості аналізу	Покращення роботи системи за рахунок зменшення часу на виконання аналізу.	Аналіз роботи системи, пошук вузьких місць, модифікація алгоритмів та процесу роботи.

Продовження Таблиці 4.7

1	2	3	4
2	Отримання зворотнього зв’язку від користувача	Покращення роботи системи в тих місцях, де користувачу найбільш незручно. Зменшення часу на доробку частин системи з якими немає проблем.	Створення сервісу для збору зворотнього зв’язку. Постійна комунікація з цільовою аудиторією.
3	Розширення функціоналу системи	Додання нового функціоналу в системі. Додання функціоналу який є не особливо важливим для системи, але є необхідним для задоволення всіх потреба користувача.	Створення плану по розвитку системи. Розробка нових модулів, що збільшують зручність користування.

Потрібно провести аналіз поточного ринку, та конкуренції яка на ньому існує для створення найкращої протидії (таблиця 4.8). Якщо бути підготовленим до конкуренції, та мати стратегії старту проекту, можна дуже збільшити його успіх.

Таблиця 4.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	**В чому проявляється дана характеристика**	**Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)**
1	**2**	**3**
1. Тип конкуренції - монополія	Кожен з великих конкурентів зайняв свою нішу та розвиває продукт з огляду не головну ідею продукту.	Розширяти вплив за рахунок тісної співпраці з цльовою аудиторією.
2. За рівнем конкурентної боротьби - локальна	Локальна, так як продукт в основному націлений на англомовні країни.	Конкуренти є інтернаціональними, за рахунок цього втрачають свої сили на локальному рівні. Потрібно розширяти ринок споживачів поступово, для старту обрати одну країну.
3. За галузевою ознакою - внутрішньогалузева	Внутрішньогалузева, тому що всі конкуренти зосереджені на аналізу даних про поведінку користувача або їх груп.	Слідкування за тендеціями у розвитку всіх соціальних мереж, при необхідності зміна ринку.

Продовження Таблиці 4.8

1	2	3
4.За характером конкурентних переваг – нецінова	Цінова, так як здебільшого конкуренти вимагають плату за проведення певного аналізу	Створення сприятливих умов для початку користування системою. Надання можливості спробувати систему перед тим як почати користуватись нею.
5. За інтенсивністю не марочна	Не марочна так як бренд не важливий у просуванні системи. Єдиною важливою частиною системи є тип аналізу та його якість.	Надати високу якість аналіз, знайти свою цільову аудиторію.

Можемо побачити що деяка конкуренція присутня (таблиця 4.9-4.10). Але прямих конкурентів які значно б ускладнили вихід на ринок немає, а тому впровадження системи має пройти на високому рівні без виникнення проблем.

Таблиця 4.9 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	**Прямі конкуренти в галузі**	**Потенційні конкуренти**	**Постачальники**	**Клієнти**	**Товари-замінники**
	Audiense	З аналізу ринку можна побачити що немає на даному етапі особливих потенційних клієнтів, які б заважали виходу на ринок	Немає необхідності	Переважно маркетологи та рекламодавці	Відсутні
Висновки:	Відсутні прямі конкуренти	Ймовірність появи потенційних конкурентів є дуже низькою	Не впливають ніяк на те що відбувається на ринку	Задоволення потреб клієнту на високому рівні є головною ціллю.	Товарів замінників немає, так як кожен націлений на свій тип аналізу

Таблиця 4.10 – Обґрунтування факторів конкурентоспроможності

№ п/п	**Фактор конкурентоспроможності**	**Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)**
1	**2**	**3**
1	Пришвидшення якості та швидкості аналізу	З кожним проведеним аналізом систему кешує отримані дані для можливості подальшого використаня.

Продовження Таблиці 4.10

1	2	3
2	Безкоштовність випробування системи на реальних даних для користувача	Система надає можливість користувачу перед тим як платити за аналіз, спробувати проаналізувати якісь дані безкоштовно. Система розширяє свою базу користувачів Twitter, при цьому користувач системи отримує безкоштовний аналіз.
3	Можливість отримання груп користувачів схожих за певними інтересами	Користувач отримує набір користувачів поєданних у групи, і вже може взаємодіяти з цими даними як йому зручно, в цьому є перевага адже клієнту надається максимальна свобода у роботі з даними.
4	Система спроектована бути легко розширяємою	З самого початку розробки системи було присвячено багато часу для того, щоб зробити систему максимально гнучкою. Саме це надає можливість легко додавати новий функціонал до системи. Створювати нові модулі. Покращувати існуючий функціонал, не впливаючи на інші частини системи. Робить процес помилка-виправлення максимально швидким.

За факторами конкурентоспроможності, проведено аналіз наявних конкурентів, в цьому процесі було знайдено сильні та слабкі сторони системи яка розроблялася (таблиця 4.11).

Таблиця 4.11 – Порівняльний аналіз сильних та слабких сторін

№ п/п	**Фактор конкурентоспроможності**	**Бали 1-20**	**Рейтинг товарів-конкурентів у порівнянні з системою**						
			–3	–2	–1	0	+1	+2	+3
1	Пришвидшення якості та швидкості аналізу	20	+						
2	Безкоштовність випробування системи на реальних даних для користувача	18		+		+			
3	Можливість отримання груп користувачів схожих за певними інтересами	15					+		
4	Система спроектована бути легко розширяємою	20				+			

Наступним кроком проаналізувавши слабкі та сильні сторони створеного стартап-проекту, визначивши які є загрози та можливості у розвитку проекту так його майбутньому було побудовано SWOT-аналіз (таблиця 4.12), щоб виділити зовнішні та внутрішні фактори, які можуть якимось чином вплинути на вихід проекту на ринок, та його подальший розвиток.

Таблиця 4.12 – SWOT- аналіз стартап-проекту

Внутрішні фактори	
Сильні сторони	**Слабкі сторони**
Постійне пришвидшення якості та швидкості аналізу Безкоштовність випробування системи на реальних даних для користувача Можливість отримання груп користувачів схожих за певними інтересами Система спроектована бути легко розширяємою	Низька якість дизайну Недостаток у фінансуванні на старті проекту
Зовнішні фактори	
Можливості	**Загрози**
Покращення системи на основі зворотнього зв’язку Розширення системи Розширення бази проаналізованих користувачів Twitter	Конкуренти є монополістами на ринку, вже заявили про себе і мають гарну репутацію Недостатня зацікавленість проектом цільовою аудиторією Зміна політики надання даних зі сторони Twitter

Проаналізувавши попередні аналізи, можна спланувати заходи за допомогою який можна пришвидшити вихід на ринок не вплинувши особливо на якість системи. Комплекс альтернатив впровадження наведено в таблиці 4.13.

Таблиця 4.13 – Альтернативи ринкового впровадження стартап-проекту

№ п/п	**Альтернатива (орієнтовний комплекс заходів) ринкової поведінки**	**Ймовірність отримання ресурсів**	**Строки реалізації**
1	Виведення системи на ринок з існуючим дизайном який може не задовільняти всім стандартам які існують, але буде повністю робочим	Пошук інвесторів, яким може сподобатись ідея системи	2 місяці
2	Покращення систему в плані дизайну та зручності користування. Додання додаткового функціоналу, що може зацікавити цільову аудиторію.	Плата користувачів за сервіс	3 місяці
3	Продаж системи компанії-конкуренту та подальша розробка під брендом конкуренту	Ресурси можуть бути отримані за рахунок продажу системи, та плати розробникам системи брендом-конкурентом	1 місяць

4.4 Розроблення ринкової стратегії проекту

Було обрано цільові групи потенційних споживачів продукту (таблиця 4.14).

Таблиця 4.14 – Вибір цільових груп потенційних споживачів

№ п/п	**Опис профілю цільової групи потенційних клієнтів**	**Готовність споживачів сприйняти продукт**	**Орієнтовний попит в межах цільової групи (сегменту)**	**Інтенсивність конкуренції в сегменті**	**Простота входу у сегмент**
1	Звичайні користувачі мережі Інтернет	Низька готовність	Середній попит	Відсутність конкуренції	Середня складність
2	Рекламодавці масштабних брендів	Висока готовність	Високий попит	Збільшення конкуренції з часом	Висока складність
3	Рекламодавці локальних брендів	Середня готовність	Середній-вище середнього	Відсутність конкуренції	Середня складність
4	Маркетологи	Дуже висока готовність	Дуже високий попит	Збільшення конкуренції з часом	Середня та нижче середнього
Цільовими групами було обрано групи 3 та 4 так як вони можуть принести найбільше позитивного.					

Для покращення процесу взаємодії з цільовими групами було створено базову стратегію розвитку проекту та стратегію конкурентної поведінки (таблиці 4.15-4.16).

Таблиця 4.15 – Визначення базової стратегії розвитку

Обрана альтернатива розвитку проекту	**Стратегія охоплення ринку**	**Ключові конкурентоспроможні позиції відповідно до обраної альтернативи**	**Базова стратегія розвитку**
Виведення системи з поточним дизайном, який може не задовільняти всім вимогам вигляду та зручності	Стратегія диференціації	Тип аналізу надає найбільш широкий варіант отриманих даних	Надання користувачей можливостей аналогів якій на ринку немає.

Таблиця 4.16 – Визначення базової стратегії конкурентної поведінки

№ п/п	**Чи є проект «першопрохідцем» на ринку?**	**Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?**	**Чи буде компанія копіювати основні характеристики товару конкурента, і які?**	**Стратегія конкурентної поведінки***
1	Частково	На початковому етапі система буде нарощувати свою базу споживачів, яким система буде зручна в базовому вигляді. В подальшому планується перехоплювати аудиторію у конкурентів за рахунок реалізації нового функціоналу.	Так, якщо це принесе велике значення у систему і буде вписуватись у концепцію.	Стратегія диференціації

Враховуючи стратегії роботи з ринком, та проаналізувавши стратегію розвитку та взаємодії з конкурентами, потрібна була бути розроблена стратегія позиціонування, яка визначає те як продукт буде позиціонуватись на ринку з врахування позицій конкурентів (таблиця 4.17).

Таблиця 4.17 – Визначення стратегії позиціонування

№ п/п	**Вимоги до товару цільової аудиторії**	**Базова стратегія розвитку**	**Ключові конкурентоспроможні позиції власного стартаппроекту**	**Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)**
1	Швидкість виконання аналізу	Стратегія спеціалізації	Оптимізація	Висока швидкість обробки заявок
2	Якість виконання аналізу	Стратегія спеціалізації	Особливість виконання аналізу	Висока точність результатів аналізу
3	Можливість різностороннього використання результатів аналізу	Стратегія спеціалазації	Зручний формат результатів аналізу	Зручна форма представлення результатів

4.5 Розроблення маркетингової програми стартап-проекту

Таблиця 4.18 – Визначення ключових переваг концепції потенційного товару

№ п/п	**Потреба**	**Вигода, яку пропонує товар**	**Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити**
1	Безпека збереження данних про користувачів	Збереження всіх внутрішніх даних на окремому захищеному сервері.	Безпека збереження даних користувача, його балансу у системі
2	Надання аналізу користувачів Twitter	Аналіз сфери застосування якого можуть дуже сильно варіюватись.	Широка сфера застосування результатів аналізу.

Таблиця 4.19 – Опис трьох рівнів моделі товару

<table>
<tr><td>Рівні товару</td><td colspan="3">Сутність та складові</td></tr>
<tr><td>I. Товар за задумом</td><td colspan="3">Веб-сервіс, який групує користувачів на основі їхньої текстової складової поведінки у соціальній мережі. Використовуючи способи класифікації тексту та порівняння схожості текстової складової поведінки між двома користувачами.</td></tr>
<tr><td rowspan="5">II. Товар у реальному виконанні</td><td>Властивості/характеристики</td><td>М/Нм</td><td>Вр/Тх /Тл/Е/Ор</td></tr>
<tr><td>1. Висока точність класифіікації користувачів
2.Зручність системи у використанні
3.Швидкість роботи програмного продукту</td><td>М</td><td>Вр/Тх/Тл</td></tr>
<tr><td colspan="3">Програма пройшла тестування, та задовільняє всім вимогам, які повинні бути присутні для виходу на ринок</td></tr>
<tr><td colspan="3">Веб-сайт який надає можливість використовувати систему</td></tr>
<tr><td colspan="3">Марка: Analyzer</td></tr>
<tr><td>III. Товар із підкріпленням</td><td colspan="3">Програмне забезпечення</td></tr>
<tr><td colspan="4">Товар буде захищено за допомогою шифрування файлів коду</td></tr>
</table>

Для того щоб коректно оцінити ціни на використання сервісу потрібно знайти певні цінові межі (таблиця 4.20). Так, як основним прибутком буде виконання аналізу заявок, ціна має обраховуватись як кількість грошей заплачених за один аналіз.

Таблиця 4.20 – Визначення меж встановлення ціни

№ п/п	**Рівень цін на товаризамінники**	**Рівень цін на товарианалоги**	**Рівень доходів цільової групи споживачів**	**Верхня та нижня межі встановлення ціни на товар/послугу**
1	100 грн / аналіз	1000 грн / клік	Середній рівень доходів	Так як система видає результати аналізу у більш сирому вигляді було вирішено значно зменшити ціну за аналіз. Також зменшення ціни допоможе привести більше споживачів у систему на початковому етапі

Сформовано систему збуту (таблиця 4.21) та концепцію маркетингових комунікацій (таблиця 4.22).

Таблиця 4.21 – Формування системи збуту

№ п/п	**Специфіка закупівельної поведінки цільових клієнтів**	**Функції збуту, які має виконувати постачальник товару**	**Глибина каналу збуту**	**Оптимальна система збуту**
1	Проведення аналізу по певній аудиторії користувачів	Відсутні	Виробник-споживач	Web-ресурс

Таблиця 4.22 – Концепція маркетингових комунікацій

Специфіка поведінки цільових клієнтів	**Канали комунікацій, якими користуються цільові клієнти**	**Ключові позиції, обрані для позиціонування**	**Завдання рекламного повідомлення**	**Концепція рекламного звернення**
Є чіткі вимоги, відкриті до нових способів аналізу даних	Веб-сайти, телефон, соціальні мережі, пряма комунікація	Аналіз користувачів у соціальних мережах	Надання користувачу переваг саме такого типу аналізу, який застосований в системі	Надання способі застосування результатів аналізу

Висновки до розділу

Сервіс націлений на виконання аналізу діяльності користувачів у соціальній мережі Twitter. Результатом аналізу є групи користувачів яких об’єднують загальні інтереси та вподобання. Функціонал застосунку дозволяє виконувати цей аналіз, при цьому зберігаючи максимальну зручність для користувача.

Так як аналіз великих даних є досить популярним, на ринку вже існує певна конкуренція з боку великих систем. Але так як дані є дуже широконаправленими, способів їх застосування і аналізу теж існує безліч. Саме тому при старті проекту сервісу буде не дуже складно зайняти свою нішу на ринку.

Для того щоб старт пройшов максимально просто, був проведений попередній аналіз ринку та основних конкурентів. Знайдено основну цільову аудиторію та сплановано яким чином буде проведено приведення нових користувачів до системи.

Знайдено основні вразливі місця при старті, та проведено роботу з того, як мінімізувати ризики провалу проекту на початкових етапах.

Отже, було проведено основну роботу з того, щоб максимально спростити вихід проекту на ринок.

ВИСНОВКИ ТА РЕКОМЕНДАЦІЇ

В даній магістерській дисертації були реалізовані алгоритми кластеризації користувачів з врахування текстової складової їх поведінки. В рамках дослідження були розглянуті існуючі методи кластеризації та порівняння векторних моделей документів. В результаті роботи було обрано алгоритм кластеризації k-means. Також було використано векторну модель представлення документу для знаходження індексу схожості між двома документами.

Вхідними даними для кластеризації є текстові складові поведінки користувачів, яких необхідно згрупувати за інтересами. Текстовою складовою поведінки користувача являється агрегована колекція всіх його повідомлень (твітів).

Було модифіковано алгоритм кластеризації. Так, замість використання стандартної оцінки відстані між елементами кластеру, було використано індекс схожості між двома користувачами. Таким чином отримано нечітку оцінку, яка досить точно відображає схожість інтересів користувачів. Зі збільшенням данних про користувача, точність оцінки росте. Такий підхід дає можливість поділити користувачів на групи, які є невідовими до початку обчислення. Мета виконання роботи була досягнена.

Було проведено дослідження про інші оцінки схожості між користувачами на основі їх поведінки. Поставлено план роботи по подальшим дослідженням в рамках тематики поділу користувачів соціальних мереж на групи.

На основі обраних моделей та методів, було реалізовано веб-сервіс, який здатний виконувати кластеризацію користувачів на групи згідно його інтересів. В процесі реалізації було застосовано найкращі практики з проектування систем enterprise рівня.

Для отримання всіх повідомлень користувача використовується відкрите Twitter API. Важливим є те, що саме у Twitter більшість інформації є відкритою, тому її обсяги є дуже великими.

В якості майбутніх напрямків покращення програмного забезпечення можуть виступати наступні пункти:

- модифікація індексу схожості між користувачами, для врахування інших їх атрибутів, які маємо змогу використати;
- аналіз інших способів порівняння текстової складової поведінки;
- аналіз методів кластеризації, які засновані на щільності.

ПЕРЕЛІК ПОСИЛАНЬ

1. Omnicore Statistic [Електронний ресурс] // Режим доступу: https://www.omnicoreagency.com/twitter-statistics/

2. David Carr How Obama Tapped Into Social Networks' Power // New York Times. 2008. pp. 10-12.

3. Twitter Statistic [Електронний ресурс] // Режим доступу: https://twitter.com/

4. E.E. Milios A Systematic Study on Document Representation and Dimensionality Reduction for Text Clustering // Dalhousie University, Halifax, Nova Scotia. 2006. pp. 5-10.

5. NLTK (Natural Language Tool Kit) Tokenization and Tagging [Електронний ресурс] // Режим доступу: http://www.bogotobogo.com/python/NLTK/tokenizationtaggingNLTK

6. L. Kaufman and P. J. Rousseeuw Finding groups in data: An introduction to cluster analysis // New York: John Wiley & Sons. March 1990. pp. 34-46.

7. J. Weissbock Using External Information for Classifying Tweets // Brazilian Conference on Classifying Tweets. 2013. pp. 7-13.

8. T. Velmurugan and T. Santhaman A Comparative Analysis between K-medoids and Fuzzy C-Means Clustering Algorithms for Statistically Distributed Data Points // Journal of Theoretical and Applied Information Technology. May 15, 2011. pp. 24-23.

9. M. Steinbach A Comparison of Document Clustering Techniques // University of Minnesota. 2014. pp. 34-58.

10. T. Velmurugan and T. Santhaman A Comparative Analysis between K-medoids and Fuzzy C-Means Clustering Algorithms for Statistically Distributed Data Points // Journal of Theoretical and Applied Information Technology. 2011. pp. 32-46.

11. L. Kaufman and P.J. Rousseeuw Finding groups in data: An introduction to cluster analysis // New York: John Wiley & Sons, Inc. 1990. pp. 21-24.

12. R. Krishnapuram A fuzzy relative of the k-medoids algorithm with application to document and snippet clustering // IEEE International Conference. 1999. pp. 15-18.

13. D. Sculley Web-Scale K-Means Clustering // WWW, Raleigh, North Carolina, USA. 2010. pp. 13-18.

14. Scikit Learn 2.3. Clustering [Електронний ресурс], Режим доступу: http://scikitlearn.org/stable/modules/clustering.html.

15. A.K. Patidar Analysis of different similarity measure functions and their impacts on shared neighbor clustering approach // Int. journal of computer application. 2012. pp. 58-64.

16. Anna Huang Similarity Measures for Text Document Clustering // Christchurch, New Zealand. 2008. pp. 12-16.

17. Булгар М.М. Кластеризація користувачів за їх інтересами / М.М. Булгар // МОДС. 2018. С. 28-29.

18. Булгар М.М. Спосіб кластеризації користувачів соціальної мережі Twitter / М.М. Булгар // ІСТУ. 2018. С. 28-32.

19. PCA и кластеризация [Електронний ресурс] // Режим доступу: https://habr.com/company/ods/blog/325654/

20. Ruby [Електронний ресурс] // Режим доступу: https://www.ruby-lang.org/en/

21. Ruby On Rails [Електронний ресурс] // Режим доступу: http://rubyonrails.org/

22. PostgreSQL [Електронний ресурс] // Режим доступу: http://www.postgresql.org/

23. ActiveAdmin [Електронний ресурс] // Режим доступу: https://activeadmin.info/

24. Sidekiq [Електронний ресурс] // Режим доступу: https://sidekiq.org/

25. HTML [Електронний ресурс] // Режим доступу: /http://htmlbook.ru/html

26. CSS [Електронний ресурс] // Режим доступу: https://htmlbook.ru/css

27. JavaScript [Електронний ресурс] // Режим доступу: https://learn.javascript.ru/

28. Domain Driven Design [Електронний ресурс] // Режим доступу: https://domainlanguage.com/ddd/

29. Active Record [Електронний ресурс] // Режим доступу: https://guides.rubyonrails.org/active_record_basics.html

30. SOLID [Електронний ресурс] // Режим доступу: https://scotch.io/bar-talk/s-o-l-i-d-the-first-five-principles-of-object-oriented-design

31. MVC [Електронний ресурс] // Режим доступу: https://www.tutorialspoint.com/design_pattern/mvc_pattern.htm

ДОДАТОК А

Графічний матеріал

Схема структурна варіантів використання

Схема структурна бази даних

Схема структурна послідовності

Математична модель

Блок-схема роботи модифікованого алгоритму

Результати досліджень ефективності методу

Схема структурна компонентів